

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra elektroniky

# **Emulátor disketové mechaniky**

## **Floppy Disk Emulator**

## Zadání diplomové práce

Student: **Bc. Matěj Hybler**  
Studijní program: N2649 Elektrotechnika  
Studijní obor: 2612T003 Aplikovaná elektronika  
Téma: **Emulátor disketové mechaniky  
Floppy Disc Emulator**  
Jazyk vypracování: čeština

### Zásady pro vypracování:

1. Proveďte literární rešerši na téma náhrady disketových mechanik v průmyslových zařízeních a osobních počítačích.
2. Popište funkci disketové mechaniky, způsoby kódování a organizaci dat na disku.
3. Navrhněte a realizujte zapojení emulátoru 3.5", 5.25" a 8" disketové jednotky s možností čtení i zápisu dat. Použijte vhodný mikrokontrolér schopný provádět kódování/dekódování dat za současného přijímání/generování všech řídicích a stavových signálů simulované mechaniky. Jako úložiště dat zvolte paměťovou kartu či USB disk.
4. Navrhněte obslužný software pro mikrokontrolér a zvolte vhodný formát pro uložení dat na paměťové kartě či USB disku.
5. Odlad'te obslužný software a ověřte funkci celého zařízení na reálném hardwaru.

### Seznam doporučené odborné literatury:

KHOL, Ivan. Diskety a disketové jednotky. *Amatérské radio: časopis pro elektroniku a amatéry*. Praha: AMARO spol. s r. o. ISSN 0322-9572.

MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMELE AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky ... Praha: BEN - technická literatura, 2003.  $\mu$ C & praxe. ISBN 80-7300-077-6.

Intel Corporation [online katalogový list]. 82077A Floppy disk controller. 1994. Dostupné z: <http://pdf1.alldatasheet.com/datasheet-pdf/view/167793/INTEL/82077AA.html>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Karel Witas, Ph.D.**


Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



---

doc. Ing. Petr Palacký, Ph.D.  
vedoucí katedry




---

prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

## **Prohlášení studenta**

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 29. dubna 2019

  
.....  
podpis studenta



## **Abstrakt**

Účelem diplomové práce je návrh a realizace emulátoru, umožňující plnohodnotnou náhradu disketové mechaniky ve stávajících zařízeních s využitím moderního paměťového média bez nutnosti zásahu do zapojení původního zařízení. Emulátor má za úkol usnadnit přenos dat mezi daným zařízením a osobním počítačem. Univerzálnost tohoto emulátoru je zajištěna možností nakonfigurovat veškeré klíčové parametry pro emulaci prostřednictvím konfiguračního souboru, uloženém na paměťové kartě spolu s ostatními uživatelskými daty.

**Klíčová slova:** Emulátor, disketová mechanika, disketa, mikrokontrolér, řadič disketové mechaniky, emulace disketové mechaniky

## **Abstract**

The purpose of this masters thesis is to design emulator, which can be used to fully substitute the original floppy disk drive without need of any major hardware change. Emulator should make exchange of data with modern PC much easier. Versatility of this emulator is achieved by possibility to configure all key parameters of emulation by using of configuration file, stored on memory card together with the rest of user data.

**Key Words:** Emulator, Floppy disk drive, Floppy disk, microcontroller, floppy disk controller, floppy disk emulation

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>9</b>
<b>Seznam obrázků</b>	<b>10</b>
<b>Seznam tabulek</b>	<b>11</b>
<b>Seznam výpisů zdrojového kódu</b>	<b>12</b>
<b>1 Úvod</b>	<b>13</b>
<b>2 Náhrada disketové mechaniky emulátorem</b>	<b>14</b>
2.1 Historie disket a disketových mechanik . . . . .	14
2.2 Výskyt disketových mechanik . . . . .	14
2.3 Způsob náhrady disketové mechaniky . . . . .	15
2.4 Emulátory dostupné na trhu . . . . .	15
<b>3 Disketová mechanika</b>	<b>18</b>
3.1 Mechanická konstrukce . . . . .	18
3.2 Záznamové médium . . . . .	18
3.3 Senzory . . . . .	18
3.4 Čtecí/zapisovací hlavy . . . . .	19
3.5 Deska elektroniky . . . . .	20
<b>4 Interface disketové mechaniky</b>	<b>21</b>
4.1 Popis signálů . . . . .	21
4.2 Časování jednotlivých signálů . . . . .	25
<b>5 Způsob záznamu dat</b>	<b>26</b>
5.1 FM kódování . . . . .	26
5.2 MFM kódování . . . . .	27
<b>6 Diskový řadič</b>	<b>28</b>
6.1 Princip funkce . . . . .	28
6.2 Konstrukce . . . . .	29
6.3 Komunikace s nadřazeným CPU . . . . .	30
<b>7 Organizace dat na disketě</b>	<b>31</b>
7.1 Strana . . . . .	31
7.2 Stopa . . . . .	31
7.3 Sektor . . . . .	31

<b>8 Organizace dat stopy</b>	<b>32</b>
8.1 Začátek stopy . . . . .	32
8.2 Hlavička sektoru . . . . .	32
8.3 Data sektoru . . . . .	33
8.4 Konec stopy . . . . .	33
8.5 CRC . . . . .	33
<b>9 Formát diskety</b>	<b>34</b>
<b>10 Koncepční řešení</b>	<b>35</b>
10.1 Konfigurační soubor . . . . .	35
10.2 Způsob ukládání dat . . . . .	36
10.3 Kódování dat . . . . .	36
10.4 Dekódování dat . . . . .	36
10.5 Reakce na vstupní signály a generování stavových signálů . . . . .	36
<b>11 Realizace kódování a dekodování dat</b>	<b>37</b>
11.1 FM kódování . . . . .	37
<b>12 Konstrukce emulátoru</b>	<b>39</b>
12.1 Mikrokontrolér . . . . .	39
12.2 Napájení . . . . .	41
12.3 Komunikace s diskovým řadičem . . . . .	41
12.4 Úložiště dat . . . . .	42
<b>13 Obslužný program</b>	<b>45</b>
13.1 Ovladače . . . . .	45
13.2 Přídavné knihovny . . . . .	46
13.3 Aktualizace firmware mikrokontroléru . . . . .	47
13.4 Vývojový diagram obslužného programu . . . . .	48
<b>14 Ověření funkce emulátoru</b>	<b>49</b>
14.1 Emulace 3,5" a 5,25" mechaniky v PC . . . . .	49
14.2 Emulace 3,5" mechaniky osciloskopu LeCroy . . . . .	52
14.3 Emulace 8" mechaniky v počítači Consul 2715 . . . . .	54
<b>15 Závěr</b>	<b>55</b>
<b>Literatura</b>	<b>56</b>
<b>Přílohy</b>	<b>57</b>

<b>A</b>	<b>Příloha</b>	<b>58</b>
A.1	Hlavní smyčka emulačního programu . . . . .	58
A.2	Příklad úplného nastavení v souboru floppy.cfg . . . . .	61
A.3	Popis parametrů konfiguračního souboru . . . . .	62
<b>B</b>	<b>Příloha</b>	<b>63</b>
B.1	Seznam použitých součástí . . . . .	63

## Seznam použitých zkratk a symbolů

FDC	– Floppy Disk Controller
MCU	– Mikrokontroler
TTL	– Tranzistor-tranzistor logika
FM	– Frekvenční modulace
MFM	– Modifikovaná frekvenční modulace
SD	– Secure Digital karta
JSON	– JavaScript Object Notation
SPI	– Serial Peripheral Interface
MISO	– Master In Slave Out
MOSI	– Master Out Slave In
CD	– Card Detect
CLK	– Hodinový signál
PWM	– Pulse Width Modulation
UART	– Universal Asynchronous Receiver and Transmitter
USB	– Universal Serial Bus
LAN	– Local Area Network
DMA	– Direct Memory Access
CPLD	– Complex Programmable Logic Device
SRAM	– Static Random Access Memory
CRC	– Cyklický redundantní součet
IR	– Infra červená
VCO	– Voltage controlled oscillator
PLL	– Phase Lock Loop
MIDI	– Musical Instrument Digital Interface
VLSI	– Very-large-scale integration
PVC	– Polyvinylchlorid

## Seznam obrázků

1	Disketové mechaniky a diskety . . . . .	13
2	Emulátor GoTek SFR1M44 [5] . . . . .	16
3	Emulátor FloppyFlash [6] . . . . .	16
4	Emulátor HxC USB verze [7] . . . . .	17
5	Rozměry 3,5"disketové mechaniky (údaje v mm)[12] . . . . .	18
6	Čtecí a zapisovací hlava . . . . .	20
7	Blokové schéma obvodu MC3470[10] . . . . .	20
8	Rozložení vývodů MLW konektoru[11] . . . . .	21
9	Kabel pro připojení 3,5" a 5,25" mechanik . . . . .	21
10	Signál INDEX při 300 ot/min . . . . .	22
11	Signál čtených dat RDATA . . . . .	23
12	Průběh a závislost signálů WGate A WDate . . . . .	24
13	Časování vstupních a výstupních signálů . . . . .	25
14	Princip FM kódování [3] . . . . .	26
15	Princip MFM kódování [3] . . . . .	27
16	Princip výpočtu CRC[15] . . . . .	29
17	Řadič RPD-1 československé výroby . . . . .	29
18	Diskový řadič s VLSI obvodem DP8473 . . . . .	30
19	Blokové schéma komunikace diskového řadiče[9] . . . . .	30
20	Organizace dat na disku . . . . .	31
21	Využití budičů 74LS07 . . . . .	42
22	Prototyp emulátoru . . . . .	43
23	Schéma zapojení . . . . .	44
24	Vývojový diagram programu . . . . .	48
25	Nesprávné časování signálu /RDATA . . . . .	49
26	Správné časování signálu /RDATA . . . . .	50
27	Nesprávné časování signálu /INDEX při zápisu dat . . . . .	51
28	Opravené časování signálu /INDEX při zápisu dat . . . . .	51
29	Testovací program CheckIt . . . . .	52
30	Náhrada mechaniky osciloskopu LeCroy . . . . .	53
31	Průběh signálu STEP pro (a)R = 1,5 kΩ, (b)R = 150 Ω . . . . .	53
32	Počítač Consul 2715 . . . . .	54

## Seznam tabulek

1	Vstupní a výstupní signály disketové mechaniky . . . . .	22
2	Hodnoty časových údajů pro signál RDDATA . . . . .	23
3	Hodnoty časových údajů pro signál WDATE . . . . .	24
4	Data hlavičky datového sektoru . . . . .	32
5	Organizace datového sektoru . . . . .	33
6	Seznam nejběžnějších formátů disket [8] . . . . .	34
7	Tabulka kódování FM . . . . .	37
8	Tabulka dekódování FM . . . . .	37
9	Tabulka kódování MFM . . . . .	38
10	Tabulka dekódování MFM . . . . .	38
11	Přehled vhodných mikrokontrolerů [2] . . . . .	41
12	Hodnoty registru pro konkrétní hodnoty periody . . . . .	45
13	Popis parametrů - X nahradit "A"nebo "B" . . . . .	62
14	Seznam součástek . . . . .	63

## Seznam výpisů zdrojového kódu

1	Příklad konfiguračního souboru . . . . .	35
2	Hlavní programová smyčka . . . . .	58
3	Příklad kompletního nastavení emulovaných jednotek souboru floppy.cfg . . . . .	61



# 1 Úvod

Velké množství průmyslových či měřicích zařízení vyrobených v době ještě před masovým rozšířením moderních záznamových médií typu Flash, byla vybavena disketovými mechanikami umožňující ukládání dat a jejich přenos mezi počítači. Disketa byla ve své době velmi hojně používané přepisovatelné paměťové médium, hlavně díky své poměrně nízké ceně a celkové rozšířenosti. Stala se tak standardem pro jednoduchý způsob přenosu dat mezi zařízeními. V dnešní době se však jedná o zastaralou a neperspektivní technologii s poměrně nízkou kapacitou, malou životností a poruchovostí.

Zařízení vybavené disketovou mechanikou jako jediným možným způsobem přenosu dat je dnes často odsouzeno k vyřazení i přesto, že jeho ostatní vlastnosti stále splňují požadavky na jeho provoz. Z ekonomických důvodů tedy může být výhodné zastaralé zařízení vylepšit výměnou disketové mechaniky za její emulátor, který plnohodnotně emuluje chování a signály generované původní mechanikou. Kromě odpojení původní mechaniky a připojení emulátoru na její místo tedy není zapotřebí žádného dalšího zásahu, ať už na straně hardwaru tak i softwaru.

Emulátory disketových mechanik jsou již dnes běžně v prodeji. Jejich velkou nevýhodou je však nemožnost plné konfigurace za provozu a jejich přizpůsobení danému zařízení bez úpravy původního firmware. Emulátor je výrobcem často pevně nakonfigurován na určitý standard a není možné jej použít s jiným zařízením.

Cílem této práce je vytvořit univerzální emulátor pro náhradu jakékoliv disketové mechaniky využívající standardní interface. Zařízení bude umožňovat současné emulování dvou disketových jednotek zároveň. To je užitečné především u historických počítačů, kde se systém zavádí z jedné diskety a na druhé jsou pak uložena ostatní data. Uživatel bude mít navíc možnost si pomocí parametrů specifikovaných v konfiguračním souboru umístěném spolu s daty na paměťové kartě nastavit emulátor dle svých potřeb. Pro případné další rozšíření funkcí bude umožněno provést update firmware emulátoru přes SD kartu. Takto všestranně nastavitelný emulátor ocenní především sběratelé historické výpočetní a měřicí techniky, kde by bylo neustále přeprogramování firmware pro přizpůsobení emulátoru různým zařízením velmi nepraktické.



Obrázek 1: Disketové mechaniky a diskety

## 2 Náhrada disketové mechaniky emulátorem

Náhrada disketové mechaniky pomocí emulátoru představuje perspektivní způsob vylepšení starších zařízení, které používají tento již zastaralý typ paměťového média. Přínos je především ve zvýšení spolehlivosti a jednoduchosti přenosu dat do novějších zařízení a současné výpočetní techniky.

### 2.1 Historie disket a disketových mechanik

První disketa a disketová mechanika byla vyvinuta koncem šedesátých let firmou IBM. Měla sloužit jako náhrada dosud používaných děrných štítků a datových pásek u sálových počítačů, které uchovávaly pouze malé množství informací čtení dat bylo sekvenční, tudíž velice pomalé. Data byla na disketu nahrána již při výrobě a bylo možné je pouze číst. Vzhledem k rotujícímu médiu a pohyblivé hlavě bylo možné velmi rychle přistoupit na kterékoliv místo na disku v poměrně krátké době, což znamenalo velký pokrok. V roce 1973 firma IBM vydala tzv. "Type 1 Diskette" o velikosti 8 palců (200 mm) v průměru, která se velmi rychle rozšířila a stala standardem až do počátku osmdesátých let, kdy byla nahrazena 5,25 palcovou disketou a později, v druhé polovině osmdesátých let, pak 3,5 palcovou disketou vyvinutou japonskou firmou Sony.

### 2.2 Výskyt disketových mechanik

Disketová mechanika se vyskytuje především u zařízení vyrobených před rokem 2000. Vzhledem k rozvoji spolehlivějších elektronických pamětí typu flash s mnohonásobně vyšší kapacitou a menšími rozměry se postupně přestala do nově vyráběných zařízení integrovat.

- **Osobní počítače**

S disketovou mechanikou je možné se běžně setkat u klasických stolních či přenosných počítačů vyrobených až do roku 2010. Před masovým rozšířením sběrnice USB a internetového připojení, představovala nejjednodušší způsob přenosu dat mezi počítači. První PC, vyrobené firmou IBM, disponovalo dvěma 5,25" disketovými mechanikami jako jediným možným způsobem uložení dat. Kromě uživatelských dat musely obsahovat i operační systém nutný pro spuštění počítače. Disketové jednotky je však možné vidět i u 8-bitových počítačů z konce sedmdesátých let. Vzhledem k tehdejší vysoké ceně nebyly tak často využívány a řada uživatelů dala přednost ukládání dat na pomalé ale cenově dostupné audio kazety.

- **CNC zařízení a ostatní průmyslové stroje**

Velké množství průmyslových zařízení využívalo data uložená na disketě jako zdroj informací pro svou činnost. V textilním průmyslu se jednalo o různé pletací stroje či například stroje vytvářející výšivky na oblečení. Ve strojním průmyslu pak různá CNC zařízení v podobě automatických fréz a soustruhů.

- **Měřicí technika**

Řada profesionálních digitálních osciloskopů, spektrálních či logických analyzátorů před zavedením a rozšířením USB sběrnice používala disketu pro přenos naměřených dat do PC.

- **Audio technika**

V hudebním průmyslu je možné disketové mechaniky spatřit v zařízeních a hudebních nástrojích podporující MIDI (Musical Instrument Digital Interface). Jsou na nich uloženy zvuky hudebních nástrojů.

## **2.3 Způsob náhrady disketové mechaniky**

Disketová mechanika je v zařízení běžně připojena přes standardní rozhraní, společné pro všechny typy disketových mechanik. Fyzické propojení je v případě 3,5" jednotek realizováno dvouřadovým 34-pinovým konektorem typu MLW. 5,25" a 8" mechaniky pak používají nožový konektor, který se nasazuje přímo na plošný spoj disketové mechaniky. Napájení je pak realizováno pomocí 4-pinového konektoru typu Molex. Pokud má být zajištěna jednoduchá zaměnitelnost emulátoru za mechaniku bez nutných dodatečných úprav, je nutné aby, emulátor disponoval právě tímto typem konektorů. Rozměr emulátoru by měl být stejný jako klasická disketová mechanika, aby bylo možné využít k uchycení původních montážních otvorů v zařízení.

## **2.4 Emulátory dostupné na trhu**

Na trhu je dostupných několik typů emulátorů disketové mechaniky. Jedná se však většinou o zařízení podporující pouze standardní disketové formáty (formát typu DOS), které používá standardní PC a umí emulovat vždy pouze jeden typ disketové mechaniky. Zde je uvedeno několik zástupců, které je dnes možné zakoupit:

### **2.4.1 GoTek SFR144MB, SFR12MB, SFR720kB**

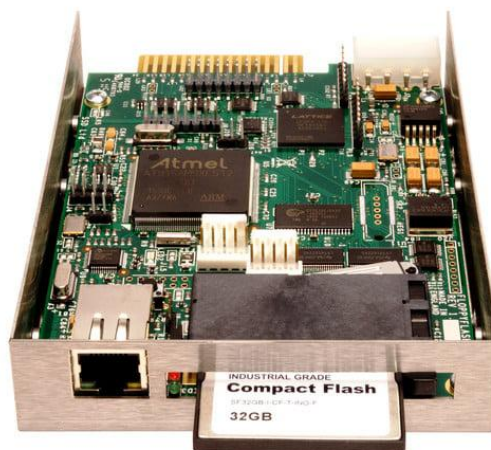
Jedná se o produkty čínského výrobce. Tento emulátor disketové mechaniky má standardní rozměry 3,5" disketové mechaniky. Jeho využití je převážně v průmyslovém, oděvním nebo dřevozpracujícím odvětví. Není vhodný pro použití v hudebních nástrojích. Pro připojení je osazen standardním konektorem typu 34 pin IDC (Male), který přenáší pouze data. K napájení je emulátor vybaven zvlášť konektorem Molex 4 pin. Oba kabely by měly být součástí zařízení, v kterém se bude mechanika vyměňovat. Tento typ emulátoru není osazen segmentovým panelem. Aby byla volba emulace jednodušší, podporuje pouze emulaci jedné diskety na USB Flash disku. V praxi to znamená, že co jeden flash disk, to jedna disketa. Proto není potřeba žádný software pro formátování (nepodporuje formáty Non-Dos tj. 1.68 MB, 1.72 MB, 1.2 MB, 820 kB, 880 kB, 360 KB, 320 kB, 180 kB, 160 kB) a natažení dat na USB Flash disk (data se přepokopírují do kořenového adresáře USB flash disku). Jeho cena se pohybuje okolo 1 200 Kč[5]



Obrázek 2: Emulátor GoTek SFR1M44 [5]

#### 2.4.2 FloppyFlash Emulator

Jedná se o poměrně komplikované zařízení britského výrobce průmyslových zařízení SolidStateDisks, které vyžaduje připojení do LAN sítě. Data jsou uložena na kartě typu CompactFlash ve specifickém formátu. Nahrávání dat a konfigurace zařízení probíhá výhradně vzdáleně právě přes LAN připojení. Cenu tohoto zařízení výrobce neuvádí. Specializuje se pouze na zakázkovou výrobu.[6]



Obrázek 3: Emulátor FloppyFlash [6]

### 2.4.3 HxC Floppy Emulator

Tento emulátor vznikl jako projekt s cílem nahradit nefunkční disketovou mechaniku v historickém počítači Atari ST. První verze emulátoru s mikrokontrolérem Microchip PIC18F4620 a SD kartou sloužila pouze jako náhrada mechaniky v již zmíněném Atari ST. Později se podpora rozšířila i o ostatní formáty a mechaniky různých typů. Aktuální USB verze již používá místo mikrokontroléru obvod CPLD ALTERA 7128SLCC a externí 32 kB paměť SRAM, která slouží jako buffer pro data. Jedná se o open-source projekt s poměrně velkým počtem přispívajících programátorů. Zakoupit jej je možné na stránkách věnovaných historickým počítačům, kde výrobu zajišťuje místní početná komunita. Z tohoto důvodu existuje celá řada verzí tohoto zařízení s různými úpravami. Na oficiálních stránkách lze nalézt zdrojový kód pro CPLD obvod v jazyce VHDL. Zdrojový kód pro SD verzi s mikrokontrolérem PIC není volně ke stažení, dostupný je pouze finální zkompileovaný firmware, který je do obvodu potřeba naprogramovat. Cena se pohybuje v přepočtu okolo 2 000 Kč. [7]

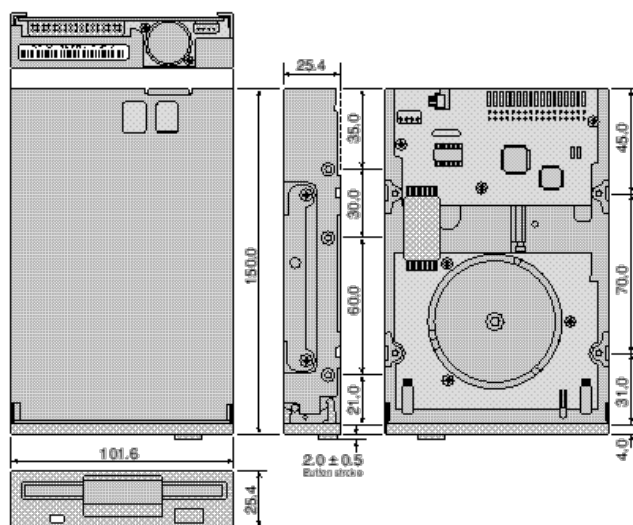


Obrázek 4: Emulátor HxC USB verze [7]

### 3 Disketová mechanika

#### 3.1 Mechanická konstrukce

Mechanická část zajišťuje po vsunutí diskety její aretaci v pracovní poloze. Točivý moment na rotující disk je z motoru přenášen u starších provedení řemenem (8" mechaniky i první 5,25") nebo je motor přímo součástí upínacího mechanismu diskety (novější 5,25" a všechny 3,5"). Vystavovací mechanismus čtecích hlav pohybujících se na kolejnicích v podélném směru je řešen motorem se šnekovým nebo planžetovým převodem.



Obrázek 5: Rozměry 3,5" disketové mechaniky (údaje v mm)[12]

#### 3.2 Záznamové médium

Disk tvoří kotouč z polyethylenu s nanesenou vrstvou oxidu železitého, železno-železitého či barnato-železitého v závislosti na požadované hustotě záznamu. Proti mechanickému namáhání a poškození je tento kotouč umístěn v obálce vyrobené z PVC plastu, která je u 8" a 5,25" disket pružná, u 3,5" disket pak pevná.

#### 3.3 Senzory

Senzory uvnitř disketové mechaniky poskytují informaci, zda je v mechanismu vložena disketa, zda je chráněna proti zápisu, jaká je její kapacita, dále pak o aktuální poloze čtecí hlavy a rotujícího disku.

### **3.3.1 Senzor vloženého disku a zamezení zápisu**

Po vložení diskety do upínacího mechanismu dojde k aktivaci senzoru přítomnosti vloženého záznamového média, případně uzamčení možnosti zápisu na disketu. Senzor je obvykle řešen mikropínačem s nízkým zdvihem a silou sepnutí. Řada mechanik však využívá optického snímání s využitím optozávory. Paprsek vysílaný infračervenou diodou je zastíněn před příchodem do fototranzistoru právě vloženou disketou nebo posuvným přepínačem v případě zámku zápisu na disketu.

### **3.3.2 Senzor záznamové hustoty**

Do disketové mechaniky je možné vložit diskety s různou hustotou záznamu. Aby bylo možné diskety s různou záznamovou hustotou odlišit, jsou vybaveny otvorem na specifickém místě. V případě 3,5" diskety je tento snímač řešen stejným způsobem jako v případě senzoru vloženého disku. U 8" a 5,25" disket je jeho funkce sdružena s indexovým senzorem.

### **3.3.3 Senzor indexu**

Vzhledem k tomu, že jsou data na disketu uložena v soustředných kruzích, je potřeba určit, na kterém místě se nachází první datový sektor. Tento senzor generuje krátký pulz při každé jedné otáčce disku o 360°. Pro diskety o rozměrech 8" a 5,25" je tvořen malým otvorem přímo v magnetickém disku, který je snímán výhradně optickou cestou pomocí IR diody a fototranzistoru. U 3,5" mechanik s přímým náhonem tuto funkci zajišťuje snímač polohy hřídele na principu Hallova snímače.

### **3.3.4 Sensory polohy hlavy**

Pro určení výchozí polohy čtecí hlavy je použit senzor, který se aktivuje po vystavení hlavy na stopu číslo 0. Informaci o aktuální poloze hlavy si uchovává sám řadič podle počtu kroků vykonaných motorem od výchozí polohy. U starších 8" mechanik je přítomen přídavný senzor vystavení hlav na stopu číslo 34, který způsobí snížení záznamového proudu při přechodu na stopy nižší než 34. Proměnnou hodnotou záznamového proudu se kompenzuje větší obvodová rychlost na vnějších stopách disku. S rychlostí průchodu zmagnetizovaného povrchu disku přes čtecí hlavu roste i napětí indukované v jejím vinutí.

## **3.4 Čtecí/zapisovací hlavy**

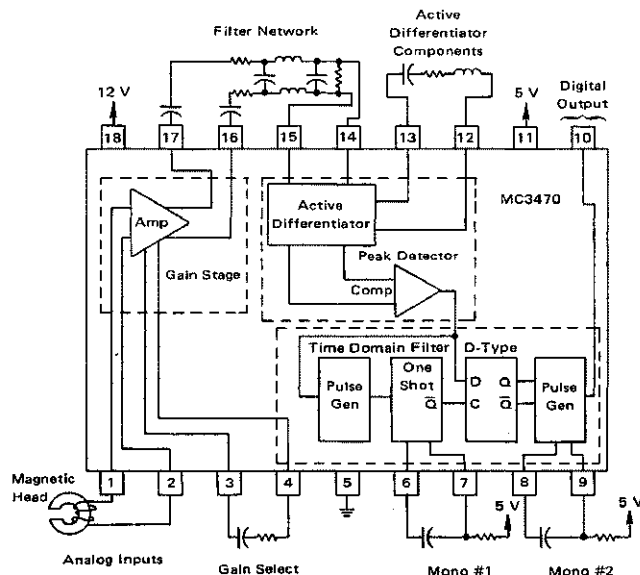
Hlavy jsou v přímém kontaktu s povrchem magnetického disku a slouží v případě čtení dat ke snímání, v případě zápisu pak vytváření magnetizace na povrchu disku. Průchodem hlavy nad různě zmagnetizovaným povrchem dochází k indukování nízkého elektrického napětí, které je s pomocí čtecího zesilovače zesíleno na požadovanou úroveň.



Obrázek 6: Čtecí a zapisovací hlava

### 3.5 Deska elektroniky

Plní funkci řízení veškerých akčních členů přítomných v mechanice. V novějších disketových mechanikách je přítomen pouze jeden specializovaný integrovaný obvod, který v sobě přímo integruje budič krokového motoru vystavování hlav, budiče motoru zajišťujícího rotaci disku, čtecích a zapisovacích zesilovačů, vstupně/výstupní interface pro řadič a logiku, která se stará o generování stavových signálů a blokování zápisu na základě informace ze senzoru uzamčení diskety. U starších mechanik, zejména 8", je tvořena TTL logickými obvody, které ovládají analogovou část budičů motorů a zpracování hodnot ze senzorů. Čtecí a záznamové zesilovače jsou často řešeny integrovaným obvodem sdružujícím zesilovač, aktivní diferenciátor, špičkový detektor a generátor pulzů o úrovních TTL, které se pak již přímo zpracovávají v diskovém řadiči. Takovým představitelem je například obvod MC3470.

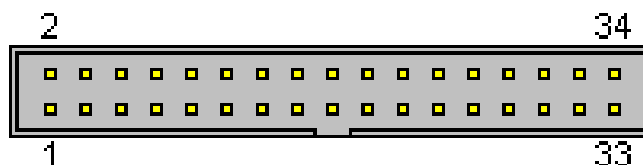


Obrázek 7: Blokové schéma obvodu MC3470[10]

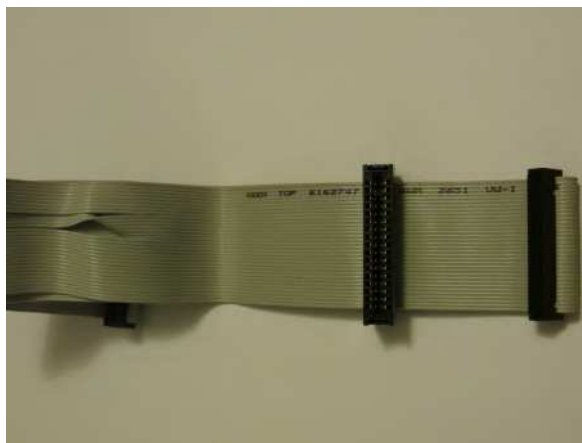


## 4 Interface disketové mechaniky

Komunikace mezi disketovou mechanikou se standardním rozhraním a řadičem probíhá pomocí 15ti signálové sběrnice. Všechny signály mají úroveň 5 V TTL logiky, jsou aktivní při nízké úrovni a kvůli impedančnímu přizpůsobení jsou vždy na straně přijímače zakončeny rezistory o hodnotě  $150\Omega$  až  $1,5\text{ k}\Omega$  připojenými na napájecí napětí. Z těchto důvodů jsou výstupní signály disketové jednotky generovány pomocí logických obvodů s otevřeným kolektorem, což umožňuje i současné připojení více disketových mechanik k jednomu řadiči.



Obrázek 8: Rozložení vývodů MLW konektoru[11]



Obrázek 9: Kabel pro připojení 3,5" a 5,25" mechanik

Veškeré liché vývody datového konektoru jsou spojeny se signálovou zemí.

### 4.1 Popis signálů

Signály využívají negativní logiku, jsou tedy aktivní vždy v úrovni L. Napěťové úrovně pro stav L jsou 0 - 0,8 V, pro stav H 2 - 5 V.

#### 4.1.1 REDWC

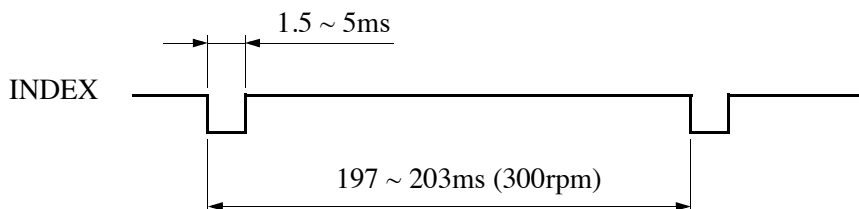
Vstupní signál disketové jednotky, který určuje, zda má být záznam proveden s nízkou nebo vysokou hustotou. Většina modernějších jednotek tento signál ignoruje a řídí se vlastním signálem ze senzoru záznamové hustoty. Signál určuje velikost zápisového proudu hlavy. Pro zápis diskety s vysokou hustotou záznamu je nutné použít vyšší proud.

Tabulka 1: Vstupní a výstupní signály disketové mechaniky

Číslo vývodu	Název	Směr	Popis
2	/REDWC	Vstup	Výběr hustoty záznamu
4	-	-	Nezapojeno
6	-	-	Nezapojeno
8	/INDEX	Výstup	Index pulzy
10	/MOTEA	Vstup	Zapnutí motoru mech. A
12	/DRVSB	Vstup	Výběr mech. B
14	/DRVSA	Vstup	Výběr mech. A
16	/MOTEB	Vstup	Zapnutí motoru mech. B
18	/DIR	Vstup	Směr pohybu hlav
20	/STEP	Vstup	Krok
22	/WDATE	Vstup	Data zápisu
24	/WGATE	Vstup	Aktivace zápisu
26	/TRK00	Výstup	Indikace stopy 0
28	/WPT	Výstup	Blokování zápisu
30	/RDATA	Výstup	Čtená data
32	/SIDE1	Vstup	Výběr hlavy 1
34	/DSKCHG	Výstup	Indikace vložené diskety

#### 4.1.2 INDEX

S každou otáčkou disku mechanika vygeneruje záporný puls trvající přibližně 1 až 8 $\mu$ s. Při rychlosti rotace disku 300 ot/min má signál periodu  $200 \pm 4$  ms, při 360ot/min pak  $166,7 \pm 2,5$  ms.



Obrázek 10: Signál INDEX při 300 ot/min

#### 4.1.3 MOTEA

Při úrovni L dojde k roztočení motoru mechaniky A.

#### 4.1.4 DRVSB

Signál výběru mechaniky B. Při úrovni L dojde k aktivaci jednotky B. Jednotka reaguje na vstupní signály a generuje odpovídající výstupní signály. Při úrovni H je jednotka B v nečinnosti a všechny její výstupní signály jsou neaktivní (úroveň H). To umožňuje druhé mechanice (A) převzetí kontroly nad sběrnici. Signály DRVSB a DRVSA tedy nikdy nesmí být aktivní současně.

#### 4.1.5 DRVSA

Signál má stejný význam jako DRVSB. Platí však pro jednotku A.

#### 4.1.6 MOTE B

Při úrovni L dojde k roztočení motoru mechaniky B.

#### 4.1.7 DIR

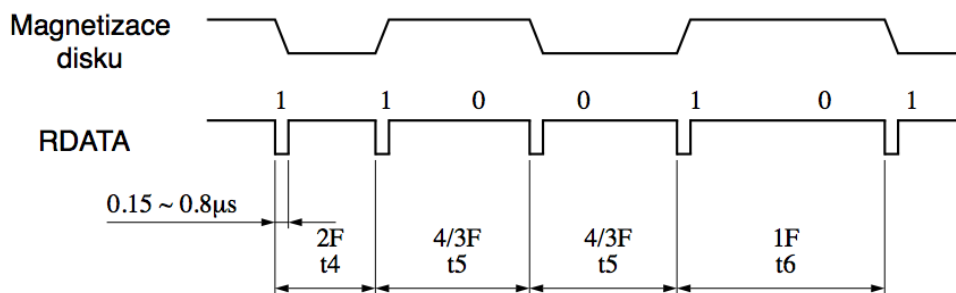
Vstupní signál určující směr pohybu diskové hlavy při krokování. Při úrovni H se hlava pohybuje k vnějšímu okraji disku (k stopě 0). V opačném případě se hlavy pohybují ke středu disku.

#### 4.1.8 STEP

Záporný puls způsobí posunutí hlavy o jeden krok směrem určeným signálem DIR. Puls musí trvat minimálně  $1\mu s$  a mezi jednotlivými pulzy je nutné dodržet  $3ms$  odstup. Doba mezi pulzy slouží pro čekání než krokový motor provede posuv hlavy mezi dvěma stopami. Dokončení přesunu hlavy na požadovanou stopu signalizují příchozí pulzy signálu INDEX, které jsou během krokování blokovány.

#### 4.1.9 RDATA

Čtená data z právě aktivní hlavy určené signálem SIDE1. Signál musí splňovat požadavky uvedené na obrázku 11. Časové údaje, které se liší pro zvolenou hustotu záznamu jsou uvedeny v tabulce 2.



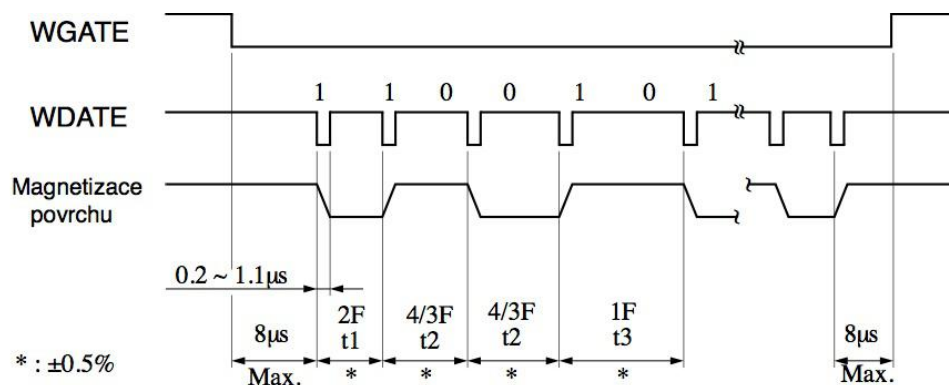
Obrázek 11: Signál čtených dat RDATA

Tabulka 2: Hodnoty časových údajů pro signál RDATA

Označení	Vysoká hustota	Nízká hustota
$t_4$	$2 \mu s$	$4 \mu s$
$t_5$	$3 \mu s$	$6 \mu s$
$t_6$	$4 \mu s$	$8 \mu s$

#### 4.1.10 WDATE

Data zapisovaná na disk. Data jsou kódována FM nebo MFM kódováním. Signál musí splňovat požadavky znázorněné na obrázku 12. Každý pulz způsobí změnu směru proudu procházející záznamovou hlavou a tím i změnu magnetizace povrchu disku.



Obrázek 12: Průběh a závislost signálů WGATE A WDATE

Tabulka 3: Hodnoty časových údajů pro signál WDATE

Označení	Vysoká hustota	Nízká hustota
t1	2 μs	4 μs
t2	3 μs	6 μs
t3	4 μs	8 μs

#### 4.1.11 WGATE

Vstupní signál indikující aktivní zápis. Při aktivaci signálu dojde k aktivaci omazávání. Datové pulzy přítomné na signálu WDATE jsou zapisovány na disk.

#### 4.1.12 TRK00

Výstupní signál, který při nízké úrovni indikuje, že se disková hlava právě nachází na stopě 0. Informace se získává ze senzoru stopy 0.

#### 4.1.13 WPT

Při úrovni L indikuje diskovému řadiči zákaz zápisu dat na disketu. Pokud je signál aktivní, záznamové obvody jednotky jsou vyřazeny z činnosti a signály WDATE a WGATE jsou ignorovány.

#### 4.1.14 SIDE1

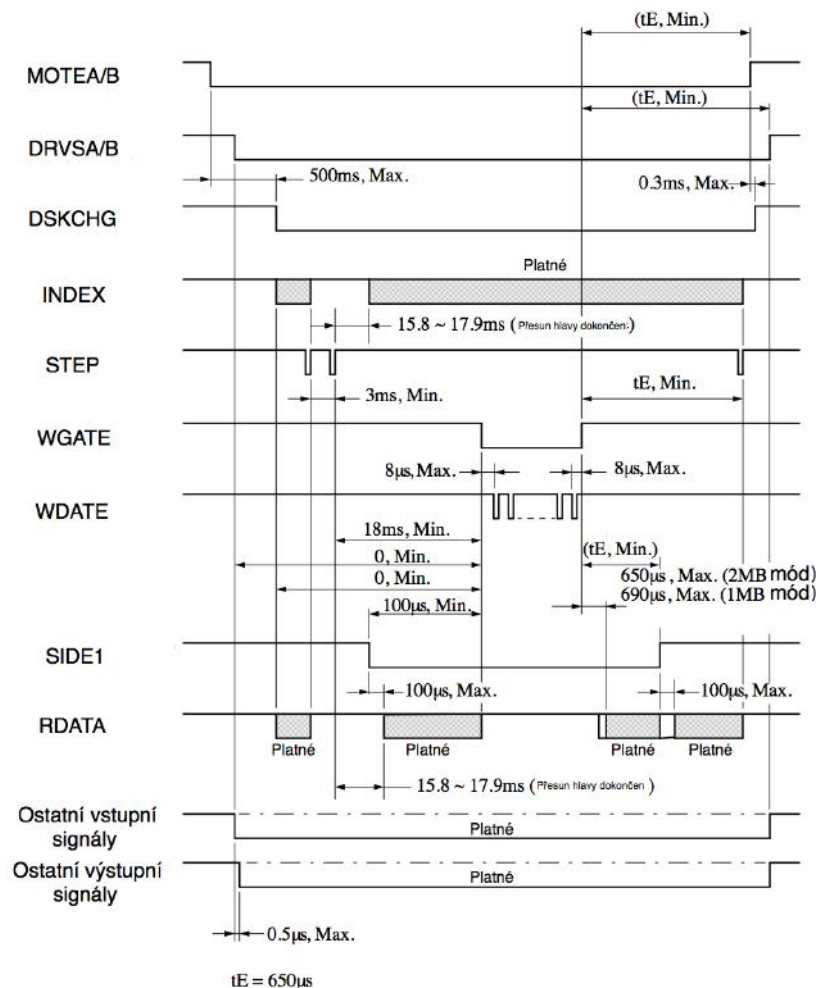
Slouží pro výběr povrchu čtených/zapisovaných dat. Pokud je signál aktivní, je vybrána hlava 1, tedy horní povrch disku. V opačném případě je aktivní hlava 0 (spodní).

#### 4.1.15 DSKCHG

Signál indikuje, zda je v mechanice vložena disketa. Při aktivaci jednotky a vložení diskety je signál aktivní.

### 4.2 Časování jednotlivých signálů

Jednotlivé signály jsou na sobě navzájem závislé a musí respektovat zásady zobrazené na obrázku 13.



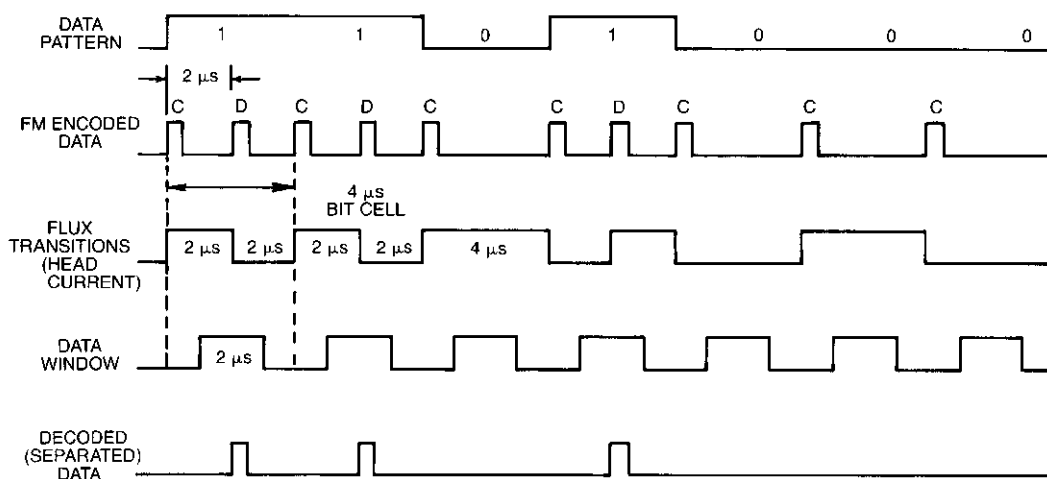
Obrázek 13: Časování vstupních a výstupních signálů

## 5 Způsob záznamu dat

Záznam dat na magnetická média z fyzikálního hlediska neumožňuje zaznamenat stejnosměrnou složku, protože elektrické napětí se v cívce čtecí hlavy indukuje pouze při změně magnetického pole. Tato změna je vyvolaná průchodem rotujícího disku s různě zmagnetizovaným povrchem. V případě nutnosti zaznamenat na disk data s několika po sobě jdoucími stejnými bity by došlo ke ztrátě informace. Je proto třeba zvolit vhodné kódování dat tak, aby bylo možné spolehlivě zaznamenat a přečíst každý samostatný bit bez ztráty dat a synchronizace. Při rotaci disku rychlostí 300 ot/min je potřeba aby došlo ke změně směru magnetizace povrchu nejpozději každých 8  $\mu\text{s}$  při záznamu dat s nízkou hustotou záznamu a 4  $\mu\text{s}$  při záznamu s vysokou hustotou záznamu. Data jsou pak na disku zaznamenána sledem různě vzdálených pulzů, představujících reverzaci magnetického pole na povrchu disku. Reverzace magnetického pole se při záznamu dat na disk dosáhne změnou směru proudu procházejícím cívkou čtecí zapisovací hlavy.

### 5.1 FM kódování

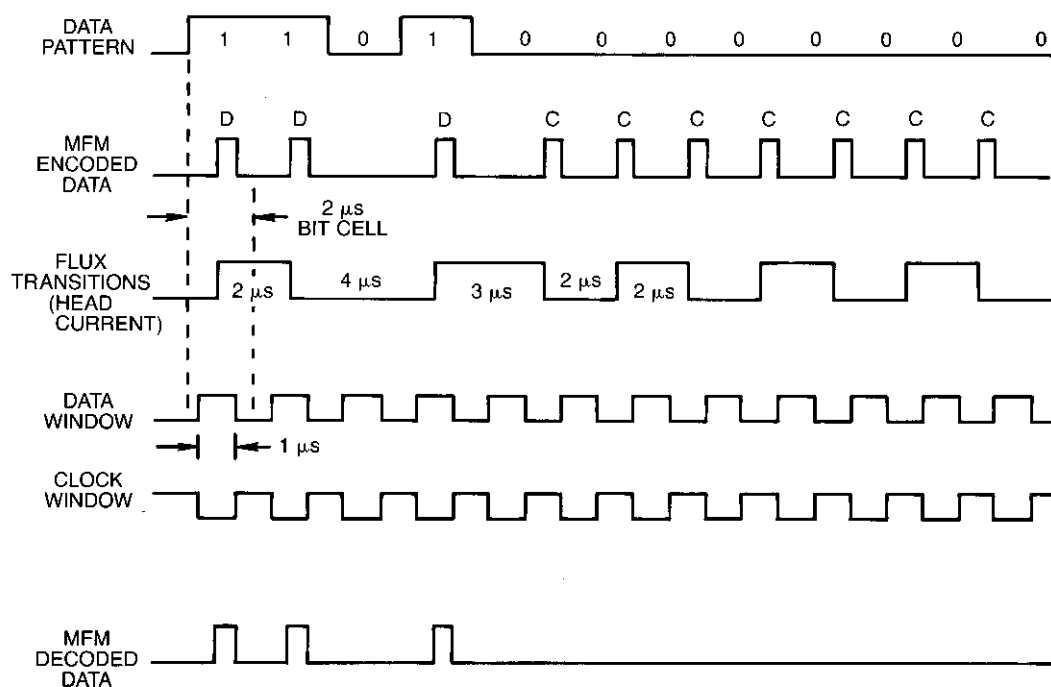
Frekvenční modulace byla prvním používaným způsobem modulace při zápisu dat na disk. Každý jednotlivý bit dat je obestoupen dvěma hodinovými bity, které slouží pro synchronizaci a zajišťují potřebnou reverzaci směru magnetizace každé 4  $\mu\text{s}$  nebo každých 8  $\mu\text{s}$ . Mezi hodinovými pulzy je tzv. okno pro jednotlivé datové bity. V případě, že je datový bit log. 1, objeví se v okně pulz. V opačném případě je mezi hodinovými pulzy mezer. Velkou nevýhodou FM kódování je, že až polovinu pulzů tvoří právě redundantní hodinové pulzy. Není tedy možné dosáhnout velké datové hustoty. Princip FM kódování je znázorněn na obrázku 14.



Obrázek 14: Princip FM kódování [3]

## 5.2 MFM kódování

Způsob kódování dat pomocí modifikované frekvenční modulace přináší oproti klasické frekvenční modulaci podstatné zvýšení hustoty zapsaných dat. Změna spočívá ve využití některých hodinových pulzů pro data. Hodinové pulzy jsou zařazeny pouze v případě záznamu několika po sobě jdoucích log. 0, aby byla opět dodržena nutná reverzace každé 4 nebo 8  $\mu\text{s}$ . Princip MFM kódování je znázorněn na obrázku 15.



Obrázek 15: Princip MFM kódování [3]

## 6 Diskový řadič

Diskový řadič je speciální obvod zajišťující komunikaci mezi disketovou jednotkou a mikroprocesorem počítače.

### 6.1 Princip funkce

Obvod řadiče je sestaven z několika dílčích funkčních bloků, které dohromady tvoří poměrně komplikovaný systém navzájem závislých jednotek.

#### 6.1.1 Obvody interface

Jsou sestaveny z budičů sběrnice s otevřeným kolektorem. Použity jsou často invertory 7406, jejich neinvertující varianta 7407, případně 2-vstupové hradla NAND 7438. Spolu se zakončovacími rezistory jsou pak tyto obvody připojeny přímo ke konektoru disketové mechaniky.

#### 6.1.2 Separátor dat

Data jsou na disketě uložená v zakódované formě buď pomocí FM nebo MFM kódování. Pro jejich úspěšné dekódování je třeba oddělit hodinové pulzy od užitečných datových. Separátor je založen na principu fázové smyčky. Rychlost rotace disku totiž nemusí být zcela konstantní a může se pohybovat v jisté toleranci. Protože je nutné, aby data uložená na disketě pomocí jedné mechaniky byla čitelná i v ostatních mechanikách, využívá se právě fázové smyčky, která se vždy adaptuje na příchozí tok kódovaných dat. Začátek diskové stopy je vždy tvořen speciální sekvencí pulzů, která zajistí správné zasynchronizování fázového závěsu.

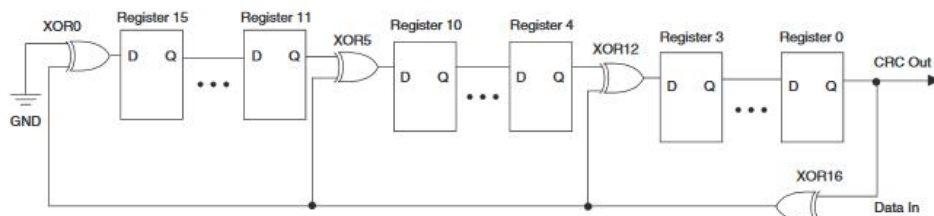
#### 6.1.3 Dekodér dat

Po zpracování dat a oddělení hodinových bitů je potřeba sériová data převést zpět na jednotlivé bajty. Dekodér je tvořen posuvnými registry, do kterých jsou příchozí bity nasouvány s hodinovým signálem ze separátoru. Postup dekódování se liší v závislosti na použitém systému kódování. Pro správné dekódování dat je potřeba s jistotou určit začátek dat. Vzhledem k tomu, že při MFM kódování hodnota právě čteného bitu závisí na hodnotě bitu předchozího, posun začátku dat o jedinný bit způsobí ztrátu synchronizace a tím i nesprávné dekódování dat. Detekce začátku dat je zajištěna použitím speciálních synchronizačních značek na začátku stopy. Kódování těchto značek vynechává hodinový pulz mezi čtvrtým a pátým bitem v jednom bajtu. To odporuje pravidlům pro kódování. Jakmile dekodér tuto speciální sekvenci detekuje, je možné s jistotou určit začátek kódovaných dat.



### 6.1.4 Výpočet CRC

Výpočet CRC kontrolního součtu realizuje série datových registrů a hradel XOR. Princip výpočtu je znázorněn na obrázku.



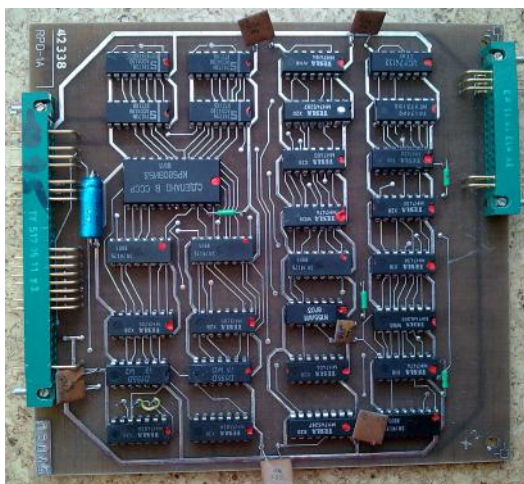
Obrázek 16: Princip výpočtu CRC[15]

### 6.1.5 Datové a konfigurační registry

Pomocí registrů je možné řadič konfigurovat a číst/zapisovat data z/na disk. Výstupní signály pro výběr mechaniky a zapnutí motoru mají svůj vlastní registr, stejně tak stavové signály o poloze čtecí hlavy nebo vložené diskety je možno číst z registru k tomu určenému.

## 6.2 Konstrukce

První řadiče disketových jednotek byly sestaveny převážně z logických obvodů řady 7400. Interface byl realizován pomocí budičů s otevřeným kolektorem. Čtená data byla přivedena do separátoru dat sestaveného z VCO oscilátoru, filtru a fázového detektoru. Pozdější verze řadičů už měly separátor realizován pomocí jediného specializovaného integrovaného obvodu. Jednotlivé stavové signály bylo možné číst a zapisovat do registrů tvořených nejčastěji obvody 74LS273.



Obrázek 17: Řadič RPD-1 československé výroby

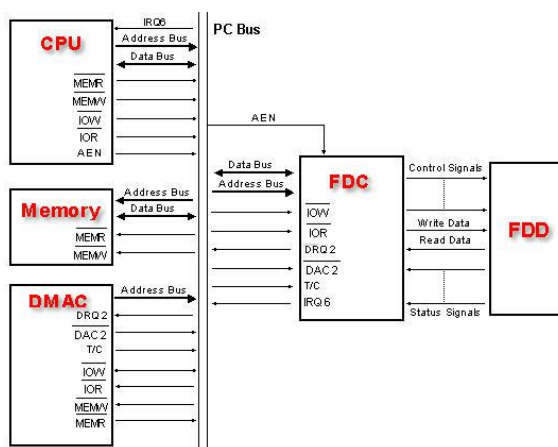
S rostoucí mírou integrace integrovaných obvodů se začaly vyrábět řadiče obsažené v jediném čipu typu VLSI. Tyto obvody v sobě obsahují kompletně řešený disketový řadič včetně separátoru a vstupně-výstupních obvodů. Mezi tyto obvody patří například Intel 82072A, NEC  $\mu$ PD765 případně jejich klony DP8473 od výrobce National Semiconductor.



Obrázek 18: Diskový řadič s VLSI obvodem DP8473

### 6.3 Komunikace s nadřazeným CPU

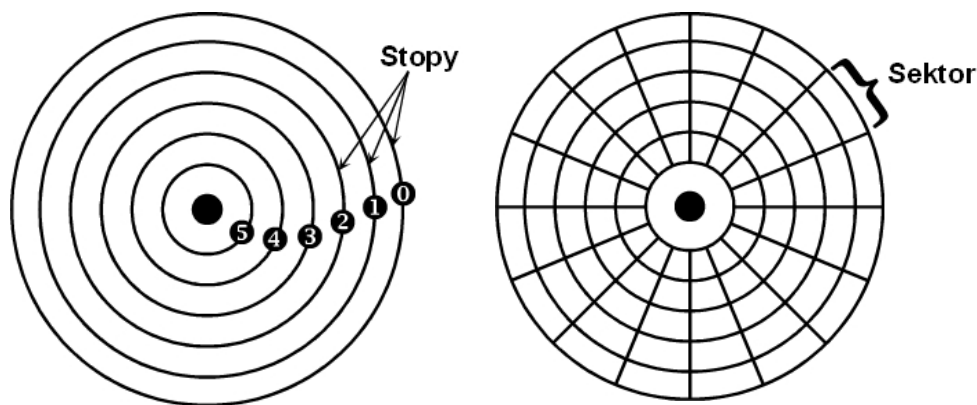
Diskovému řadiči je přiřazen jeden vektor přerušení spolu s jedním DMA kanálem. Řadič je tedy možno používat tzv. polling módu, kdy se o přenos veškerých dat mezi pamětí a řadičem stará mikroprocesor. Díky tomu ale nelze použít procesor současně k jiným operacím, takže je z hlediska využití CPU velmi neekonomický. Standardně se tedy provede nastavení DMA kanálu a o přenesení dat z/do se řadiče postará DMA jednotka. CPU je tak možné současně využívat pro jiné účely.



Obrázek 19: Blokové schéma komunikace diskového řadiče[9]

## 7 Organizace dat na disketě

Data jdou na disketě uložena po obou stranách v soustředných kružnicích, kdy každá kružnice představuje právě jednu stopu. Stopy se dále dělí na jednotlivé sektory. Organizace je znázorněna na obrázku 20.



Obrázek 20: Organizace dat na disku

### 7.1 Strana

Disketa může pro záznam dat využívat oba povrchy disku, kdy pro každý povrch je určena samostatná čtecí/zapisovací hlava. Existují však mechaniky, které jsou vybaveny pouze jedinou hlavou. Kapacita dat, které je možné na disketu uložit, je tedy poloviční. Spodní povrch disku se označuje jako strana 0, horní povrch pak jako strana 1.

### 7.2 Stopa

Stopy se číslují od nuly a nultá stopa je umístěna vždy na vnějším okraji disku, tedy nejdále od středu. Jejich počet je závislý na typu a formátu diskety.

### 7.3 Sektor

Sektor na disku představuje nejmenší možnou adresovatelnou jednotku. Je tedy možné číst, zapisovat nebo mazat vždy jeden celý sektor. Sektory jsou číslovány vždy od jedničky. Počet a velikost jednotlivých sektorů ve stopě závisí na použitém formátu disku.

## 8 Organizace dat stopy

Vzhledem k použitému typu kódování je bezpodmínečně nutné správně určit začátek dat na disku. Stopa obsahuje kromě sektorů s užitečnými daty také značné množství pomocných dat a značkovacích bajtů, které umožní diskovému řadiči se zasynchronizovat na začátek dat a identifikovat mezi sebou jednotlivé čtené sektory.

### 8.1 Začátek stopy

Začátek stopy indikuje sestupná hrana indexového signálu generovaného snímačem indexového otvoru pro 8" a 5,25" diskety a snímačem integrovaným v motoru pro 3,5" diskety. Ihned po sestupné hraně indexového signálu následuje 80 bajtů s hodnotou 0x4E. To poskytne diskovému řadiči dostatek času, aby nastavil svůj vnitřní stavový automat pro čekání na synchronizační značky. Po hodnotě 0x4E následuje 12 nulových bajtů a tři bajty s hodnotou 0xC2. Bajty 0xC2 jsou zaznamenány s chybějícím hodinovým pulzem mezi třetím a čtvrtým bitem. Tato skutečnost je účelně v rozporu s pravidly pro FM a MFM kódování a způsobí speciální sled pulzů, který při kódování užitečných dat nemůže nikdy nastat. Jakmile diskový řadič tento sled detekuje, dojde k zasynchronizování jeho datového separátoru, který nyní může rozpoznat datové a hodinové pulzy. Po třech synchronizačních 0xC2 bajtech je na stopě zaznamenán ještě jeden bajt o hodnotě 0xFC. Ten slouží jako kontrola správnosti zasynchronizování datového separátoru.[14]

### 8.2 Hlavička sektoru

Každý datový sektor má svou identifikační hlavičku, která obsahuje důležité informace o daném sektoru. Hlavička je opět uvedena synchronizačními daty skládající se z 50ti bajtů 0x4E, 12ti nulovými bajty a tří synchronizačních bajtů o hodnotě 0xA1. Bajt 0xA1 je zaznamenán s chybějícím hodinovým pulzem mezi bitem 4 a 5, čehož je opět využito k zasynchronizování datového separátoru. Pro kontrolu synchronizace je v případě hlavičky sektoru použit bajt 0xFE. Nyní následuje 6 bajtů samotné hlavičky s údaji o následujícím sektoru. Hlavička je zakončena dvou bajtovým CRC kontrolním součtem pro detekci případné chyby čtení. Tato ID hlavička předchází každá datový sektor.

Tabulka 4: Data hlavičky datového sektoru

Parametr	Počet bajtů
Bajt ověřující správnou synchronizaci 0xFE	1 bajt
Číslo stopy	1 bajt
Číslo strany disku (0x00 - 0x01)	1 bajt
Číslo sektoru (0x01 až max počet sektorů)	1 bajt
Délka sektoru	1 bajt
CRC kontrolní součet	2 bajty

### 8.3 Data sektoru

Datový sektor je uveden synchronizačními daty skládající se z 22 bajtů o hodnotě 0x4E, 6ti nulových bajtů a tří synchronizačních bajtů o hodnotě 0xA1. Bajt 0xA1 je zaznamenán s chybějícím hodinovým pulzem mezi bitem 4 a 5 pro zasynchronizování datového separátoru. Pro kontrolu synchronizace je v případě datového sektoru použit bajt 0xFB. Následují samotná data o velikosti uvedené v hlavičce sektoru zakončená kontrolním CRC součtem.[14]

Tabulka 5: Organizace datového sektoru

Parametr	Počet bajtů
Bajt ověřující správnou synchronizaci 0xFB	1 bajt
Data	x bajtů
CRC kontrolní součet	2 bajty

### 8.4 Konec stopy

Za posledním datovým sektorem stopy následuje 54 bajtů o hodnotě 0x4E, 6 nulových bajtů a tří synchronizačních bajtů o hodnotě 0xA1. Bajt 0xA1 je zaznamenán s chybějícím hodinovým pulzem mezi bitem 4 a 5, což způsobí zasynchronizování datového separátoru. Na konci stopy již není třeba synchronizaci kontrolovat. Zbytek místa je až do příchodu další sestupné hrany indexového signálu zaplněn bajty o hodnotě 0x4E.

### 8.5 CRC

CRC (Cyclic Redundancy Check) představuje kontrolní součet informací obsažených v hlavičce sektoru a samotných uživatelských dat. Slouží pro kontrolu správnosti čtení dat. Ověření správnosti CRC provádí přímo diskový řadič, kdy se vypočtená hodnota kontrolního součtu vždy porovnává s hodnotou uloženou na konci každého sektoru. Pakliže se hodnoty nerovnají, řadič se pokusí čtení zopakovat. CRC má 16 bitů ( $r=16$ ), používá standardu CCITT s počáteční hodnotou 0xFFFF využívající charakteristický polynom:

$$g(x) = x^{16} + x^{12} + x^5 + 1 \quad (1)$$

Posloupnost dat  $d$  o  $k$  bitech, ze kterých se má CRC vypočítat, je interpretována jako  $x^r d(x)$ , kde  $d = (d_0, d_1, d_{k-1})$  a jim odpovídající polynom  $d(x) = d_0 + d_1x + \dots + d_{k-1}x^{k-1}$ . Tento polynom je vydělen charakteristickým polynomem  $g(x)$ . Výsledkem je podíl a zbytek (modulo), který právě tvoří výsledek kontrolního CRC výpočtu.[13]

$$CRC = x^r d(x) \bmod g(x) \quad (2)$$

## 9 Formát diskety

Formát diskety je většinou specifický pro dané zařízení, ve kterém disketová mechanika pracuje. Formátem diskety se rozumí celkový počet stop, počet sektorů na jednotlivé stopě a jejich velikost. Měnit se případně mohou i synchronizační bajty, struktura hlaviček, méně často i rychlost rotace samotného disku. Tabulka 5 uvádí příklady různých formátů disků v závislosti na velikosti diskety, hustoty záznamu, použitém kódování a platformě.

Tabulka 6: Seznam nejběžnějších formátů disket [8]

Platforma	Rozměr	Velikost sektoru	Počet sektorů	Počet stop	Počet stran	Kapacita	Otáčky	Kódování
IBM 3740	8"	128	26	73	1	237.25 KB	360	FM
NEC PC98	8"	128	26	77	1	250.25 KB	360	FM
Apple II	5,25"	256	16	35	1	140 KB	300	MFM
Macintosh	3,5"	512	18	80	2	1440 KB	300	MFM
Atari 8-bit	5,25"	128	18	40	1	90 KB	288	FM
Atari 8-bit	5,25"	128	26	40	1	130 KB	288	MFM
Atari 8-bit	5,25"	256	18	40	1	180 KB	288	MFM
Atari	3,5"	512	9	80	1	360 KB	300	MFM
Atari	3,5"	512	9	80	2	720 KB	300	MFM
Atari	3,5"	512	18	80	2	1440 KB	300	MFM
Commodore	3,5"	512	10	80	2	800 KB	300	MFM
Amiga	5,25"	512	11	40	2	440 KB	300	MFM
Amiga	5,25"	512	11	80	2	880 KB	300	MFM
Amiga	3,5"	512	11	80	2	880 KB	300	MFM
Amiga	3,5"	512	22	80	2	1760 KB	150	MFM
IBM PC	8"	128	26	77	1	250.25 KB	360	FM
IBM PC	8"	128	26	77	2	500.5 KB	360	FM
IBM PC	8"	1024	8	77	1	616 KB	360	MFM
IBM PC	8"	1024	8	77	2	1232 KB	360	MFM
IBM PC	5,25"	512	8	40	1	160 KB	300	MFM
IBM PC	5,25"	512	8	40	2	320 KB	300	MFM
IBM PC	5,25"	512	9	40	1	180 KB	300	MFM
IBM PC	5,25"	512	9	40	2	360 KB	300	MFM
IBM PC	5,25"	512	8	80	1	320 KB	300	MFM
IBM PC	5,25"	512	8	80	2	640 KB	300	MFM
IBM PC	5,25"	512	15	80	2	1200 KB	360	MFM
IBM PC	3,5"	512	8	80	1	320 KB	300	MFM
IBM PC	3,5"	512	9	80	1	360 KB	300	MFM
IBM PC	3,5"	512	8	80	2	640 KB	300	MFM
IBM PC	3,5"	512	9	80	2	720 KB	300	MFM
IBM PC	3,5"	512	18	80	2	1440 KB	300	MFM

## 10 Koncepční řešení

Cílem této práce je vytvořit emulátor s možností emulace co nejširšího možného množství používaných disketových mechanik. Jako nejvýhodnější způsob, kterým je možné tento požadavek dosáhnout, se jeví použití konfiguračního souboru. Tento soubor bude uložen přímo na SD kartě spolu s ostatními daty. Uživatel tak bude mít možnost si emulátor nastavit přímo podle svých konkrétních požadavků na činnost tohoto přístroje.

### 10.1 Konfigurační soubor

Na paměťové kartě bude vždy v hlavním adresáři uložen konfigurační soubor `floppy.cfg`. Konfigurační data obsažená v tomto souboru mají formát strukturovaných dat. Z tohoto důvodu byl vybrán právě formát JSON, který je vhodný především pro tento typ dat. Syntaktická analýza (parsování) je pro takový formát jednoznačná a je možné v programu použít některý z řady volně dostupných parserů. Konfigurační soubor se načte při každém spuštění emulátoru a provede potřebnou konfiguraci jedné nebo obou emulovaných mechanik současně. Zde je příklad obsahu konfiguračního souboru pro emulaci dvou disketových jednotek:

---

```
{
  "pocet_mechanik": 1,
  "nazev_souboru_mech_A": "DISK_A.IMG",
  "nazev_souboru_mech_B": "DISK_B.IMG",
},
{
  "kodovani_mech_A": "MFM"
  "hustota_zazn_mech_A": "high"
  "pocet_stran_mech_A": 2
  "pocet_stop_mech_A": 80
  "pocet_sektoru_mech_A": 18
  "velikost_sektoru_mech_A": 512
},
{
  "kodovani_mech_B": "MFM"
  "hustota_zazn_mech_B": "double"
  "pocet_stran_mech_B": 2
  "pocet_stop_mech_B": 40
  "pocet_sektoru_mech_B": 9
  "velikost_sektoru_mech_B": 512
}
```

---

Výpis 1: Příklad konfiguračního souboru

## 10.2 Způsob ukládání dat

Vzhledem k tomu, že emulovaná mechanika pracuje s daty nezávisle na uloženém formátu a použitém souborovém systému, bude pro ukládání dat sloužit soubor ve formátu IMG. Jedná se o formát používaný pro tzv. obrazy disků. Binární data uložená na disketě jsou přesnou kopií dat uložených v souboru IMG. Velkou výhodou je, že se obraz disku dá na běžném osobním počítači připojit jako virtuální mechanika, se kterou je pak možné pracovat stejně jako s každým jiným úložištěm.

## 10.3 Kódování dat

Výstupní data z disketové mechaniky jsou kódována pomocí FM nebo MFM modulace. Jedná se tedy vlastně o sled pulsů s konstantní délkou, kdy se mění pouze jejich perioda opakování na základě právě přenášeného bitu. Ke generování výstupního signálu bude využit PWM modul mikrokontroléru, kde je možné jednoduchým způsobem měnit periodu výstupního signálu. Při každém pulzu se vyvolá přerušení a dojde k nastavení hodnoty periody příslušící vždy následujícímu bitu dat.

## 10.4 Dekódování dat

Příchozí tok dat z diskového řadiče je kódován stejným způsobem jako výstupní data. V tomto případě je ale nutné periodu příchozích pulzů naopak měřit. K tomuto účelu je vhodné použít jeden z dostupných modulů čítačů v mikrokontroléru. Čítač využívá jako vstup hodinový signál mikrokontroléru. S příchodem každého datového pulzu dojde k uložení hodnoty čítače, jeho vynulování a opětovné odstartování řízeného hodinovým kmitočtem mikrokontroléru. Hodnota čítače tedy vždy odpovídá časovému úseku mezi dvěma pulzy. Hodnoty časových úseků jsou následně převedeny na jednotlivé bity.

## 10.5 Reakce na vstupní signály a generování stavových signálů

Veškeré vstupní signály z diskového řadiče musí být emulátorem správně vyhodnoceny a na základě jejich stavu je třeba vytvářet odpovídající stavové signály. Mikrokontrolér umožňuje generovat přerušení při změně úrovně signálu na jeho vstupu. V obsluze přerušení je pak možné zjistit, na kterém signálu došlo ke změně a na základě vyhodnocení této informace pak emulátor nastaví výstupní signály na patřičné úrovně. Uvedený způsob představuje nejrychlejší možnou reakční dobu a způsobí nejmenší výpočetní zatěžování mikrokontroléru.



## 11 Realizace kódování a dekódování dat

Perioda, neboli vzdálenost jednotlivých pulzů, může nabývat vždy tří různých hodnot.  $4\ \mu\text{s}$ ,  $6\ \mu\text{s}$  a  $8\ \mu\text{s}$  při záznamu s nízkou hustotou nebo  $2\ \mu\text{s}$ ,  $3\ \mu\text{s}$  a  $4\ \mu\text{s}$  při záznamu s vysokou hustotou záznamu. Způsob kódování závisí na použitém způsobu modulace.

Postup při dekódování je opět třeba zvolit na základě použité modulace a hustotě záznamu. Hodnoty časových úseků mezi jednotlivými pulzy naměřené pomocí čítače je nutné vyhodnotit, převést na odpovídající hodnotu bitu a poté uložit do bufferu stopy v operační paměti.

### 11.1 FM kódování

Jednotlivé bity dat při záznamu FM jsou kódovány následujícím způsobem:

- **log. 1** - Odpovídá vzdálenosti dvou sousedních pulzů  $2/4\ \mu\text{s}$  podle hustoty záznamu.
- **log. 0** - Odpovídá vzdálenosti dvou sousedních pulzů  $4/8\ \mu\text{s}$  podle hustoty záznamu.

Tabulka 7: Tabulka kódování FM

Hodnota současného bitu	Perioda datového signálu
1	$2/4\ \mu\text{s}$
0	$4/8\ \mu\text{s}$

Dekoódoování FM datového toku se pak řídí následujícími pravidly:

- **perioda 2 nebo  $4\ \mu\text{s}$**  - Hodnota právě přenášeného bitu je 1.
- **perioda 4 nebo  $8\ \mu\text{s}$**  - Hodnota právě přenášeného bitu je 0.

Tabulka 8: Tabulka dekódování FM

Perioda datového bitu	Bit zapisovaných dat
$2/4\ \mu\text{s}$	1
$4/8\ \mu\text{s}$	0

#### 11.1.1 MFM kódování

Jednotlivé bity dat při záznamu MFM jsou kódovány následujícím způsobem:

- **log. 1 následovaná log. 1** - Pokud mají dva sousední bity stejnou hodnotu, perioda pulzu je 2 nebo  $4\ \mu\text{s}$  podle hustoty záznamu.

- **log. 0 následovaná log. 0** - Stejně jako v předchozím případě mají dva sousední bity stejnou hodnotu, perioda pulzu je 2 nebo 4  $\mu\text{s}$  podle hustoty záznamu.
- **log. 1 následovaná log. 0** - Dochází ke změně hodnoty bitu z 1 na 0. V tomto případě je perioda pulzu 3  $\mu\text{s}$  nebo 6  $\mu\text{s}$ .
- **log. 0 následovaná log. 1** - Dochází ke změně hodnoty bitu z 0 na 1. V tomto případě je perioda pulzu 4  $\mu\text{s}$  nebo 8  $\mu\text{s}$ .

Tabulka 9: Tabulka kódování MFM

Hodnota předchozího bitu	Hodnota současného bitu	Perioda datového signálu
1	1	2/4 $\mu\text{s}$
0	0	2/4 $\mu\text{s}$
1	0	3/6 $\mu\text{s}$
0	1	4/8 $\mu\text{s}$

Dekoóování MFM datového toku se pak řídí následujícími pravidly:

- **perioda 2 nebo 4  $\mu\text{s}$**  - Hodnota právě přenášeného bitu je stejná jako hodnota předchozího.
- **perioda 3 nebo 6  $\mu\text{s}$**  - Pokud byl předchozí bit 1, tento bit je 0. Pokud byl předchozí bit 0, následující dva bity jsou 01.
- **perioda 4 nebo 8  $\mu\text{s}$**  - Následující dva bity jsou vždy 01

Tabulka 10: Tabulka dekodování MFM

Perioda datového bitu	Hodnota předchozího bitu	Bit(y) zapisovaných dat
2/4 $\mu\text{s}$	1	1
2/4 $\mu\text{s}$	0	0
3/6 $\mu\text{s}$	1	0
3/6 $\mu\text{s}$	0	01
4/8 $\mu\text{s}$	1	01

## 12 Konstrukce emulátoru

### 12.1 Mikrokontrolér

Mikrokontrolér tvoří nejdůležitější součást emulátoru, který zajišťuje čtení/zápis dat na SD kartu a jejich současné kódování/dekódování do/z FM popřípadě MFM datového toku za běhu. V případě MFM kódování s vysokou hustotou záznamu je potřeba generovat datový signál o maximálním kmitočtu 500 kHz, což odpovídá periodě 2  $\mu$ s. Vzhledem k současnému načítání dat z SD karty, jejich překódování do MFM kódovaného datového toku za současného generování ostatních stavových signálů, je nutné použít mikrokontrolér se značně velkým výpočetním výkonem a odpovídající velikostí operační paměti.

#### 12.1.1 Požadavky na rychlost mikrokontroléru

Klíčovým požadavkem na použitý mikrokontrolér je jeho výpočetní výkon. Z technických specifikací pro disketové mechaniky a diskové řadiče je patrné, že časování jednotlivých signálů je velmi kritické. Nejkritičtějším parametrem je rychlost reakce na signály generované diskovým řadičem. Skutečná disketová mechanika obsahuje kromě analogové části zodpovědné za zesílení signálu z čtecích hlav převážně logické obvody, jejichž zpoždění je zanedbatelné. Veškerá zpoždění signálů a jiné prodlevy jsou dány čistě mechanickými vlastnostmi. Reakční doba mikrokontroléru je naproti tomu dána především použitou taktovací frekvencí. Z průběhu na obrázku 13 je zřejmé, že šířka pulzu signálu zapisovaných dat WRDATA se může pohybovat v rozmezí 200 až 1000 ns. Přesnost periody je  $\pm 10$  ns. Je tedy jisté, že je zapotřebí použít mikrokontrolér schopný zpracovat a generovat signály s přesností a rozlišením v řádu desítek ns. Toho je možné dosáhnout jen při taktovacích kmitočtech pohybujících se těsně pod hranicí 100 MHz.

#### 12.1.2 Požadavky na velikost operační paměti

Při kódování dat za běhu není vhodné spoléhat na přenosovou rychlost SD karty, která se může lišit mezi jednotlivými typy karet od různých výrobců. Reakce na signály od diskového řadiče také vyžadují poměrně krátkou dobu zpracování. V případě, že při současném načítání či zápisu dat na kartu začne diskový řadič požadovat změnu stopy, bylo by nutné nejprve přerušit komunikaci s kartou, aby nedošlo ke ztrátě dat a až poté reagovat na příchozí signály od řadiče. Vzhledem k poměrně kritickému časování všech signálů je tedy vyloučeno použití přímého čtení dat z paměťové karty. Pro zvýšení rychlosti zpracování a usnadnění práce s daty je výhodné načíst data pro jednu celou stopu do operační paměti. Pro kapacitu dat diskety platí:

$$\text{kapacita diskety} = (\text{velikost sektoru} \cdot \text{počet sektorů} \cdot (\text{počet stop} \cdot \text{počet stran})) \quad (3)$$

Běžná 3,5" disketa s kapacitou 1 440 kB a vysokou hustotou záznamu při 80ti stopách na dvou stranách a celkovém počtu 18 sektorů o velikosti 512 B. Pro velikost jedné stopy v tomto případě platí:

$$velikost\ stopy = (velikost\ sektoru \cdot n\ sektorů) = (512 \cdot 18) = 9\,216\ B \quad (4)$$

Při čtení či zápisu dat na disk si však diskový řadič může vyžádat změnu povrchu. Dle specifikace musí mechanika poskytnout data při změně signálu SIDE do 100  $\mu$ s. Z těchto důvodů je tedy vhodné načíst data aktuální stopy současně pro oba povrchy disku, tedy dvojnásobek velikosti jedné stopy. Minimální nutná velikost operační paměti pro načtení dat tedy činí 18,432 kB. Pro běh programu a komunikaci s kartou je třeba vyhradit dostatečné místo pro pomocná data a zajistit dostatečnou rezervu, aby nedošlo k případnému přetečení nebo kolizi.

### 12.1.3 Ostatní požadavky

Kromě početního výkonu a velikosti operační paměti je dále nutné, aby použitý mikrokontrolér disponoval důležitými periferiemi, které jsou potřebné ke komunikaci s paměťovou kartou a generováním datového signálu.

- **HSMCI / SPI sběrnice** - Paměťové karty typu SD jsou schopny komunikovat buď pomocí HSMCI interface nebo v módu SPI. Zvolený mikrokontrolér by tedy měl disponovat aspoň jednou z nich.
- **PWM** - PWM modul je nezbytnou periferií při generování datového signálu. Pro dodržení nutné přesnosti signálu je vhodné tuto činnost přenechat právě hardwaru.
- **Časovače** - Pro příjem dat z diskového řadiče a případné generování méně náročných signálů na časování bude sloužit multifunkční časovač s možností měření periody signálu.
- **GPIO** - Příchozí stavové signály z řadiče je nutné vyhodnotit, zpracovat a vygenerovat odpovídající výstupní signály s co možná nejmenším spožděním. Klíčovým parametrem je tedy rychlost zpracování přerušení a počet dostupných GPIO portů. Pro interface mechaniky je třeba celkem 15 IO signálů, pro komunikaci s SD kartou pak dalších 6 signálů. Celkem tedy minimálně 21 IO pinů.

### 12.1.4 Dostupné mikrokontroléry

Z širokého výběru vhodných a dostupných mikrokontrolérů na trhu bylo vybráno několik typů s 32bitovou architekturou, splňující výše zmíněné požadavky. Jsou uvedeny v následující tabulce:

Jako nejoptimálnější volba se jeví 32-bitový MCU ATSAM3X8E od výrobce Atmel/Microchip. Mikrokontrolér disponuje velkou datovou a operační pamětí, velkým množstvím GPIO pinů a především HSMCI rozhraním, umožňujícím vysokorychlostní komunikaci s SD kartou. MCU využívá jádro ARM Cortex-M3 taktované na 84 MHz při využití vnitřní PLL. Další předností je

Tabulka 11: Přehled vhodných mikrokontrolerů [2]

MCU	Výrobce	taktovací frekvence	Velikost paměti prog./dat	Periferie
STM32F105	ST Microelectronics	72 MHz	256 kB / 64 kB	10xTIMER, 1xPWM, 3xSPI, 80x GPIO, 1xUSB
<b>ATSAM3X8E</b>	<b>Atmel/Microchip</b>	<b>82 MHz</b>	<b>512 kB / 96 kB</b>	<b>9xTIMER, 8xPWM, 4xSPI, 103xGPIO, 1xUSB, 1xHSMCI</b>
TMS320F28035	Texas Instruments	60 MHz	64 kB / 10 kB	3xTIMER, 14xPWM, 2xSPI, 45x GPIO
TMS320F280049C	Texas Instruments	100 MHz	256 kB / 100 kB	3xTIMER, 16xPWM, 2xSPI, 40x GPIO

možnost využití USB sběrnice pro případné připojení pamětového Flash disku. Mikrokontrolér pro svou činnost vyžaduje napětí 1,8 a 3,3 V. Pouzdro je typu LQFP se 144 vývody.

### 12.1.5 Zapojení mikrokontroléru

Zapojení mikrokontroléru je provedeno na základě doporučení výrobce. Hlavní napájecí napětí 3,3V je přivedeno na vývod VDDIN a VDDIO1 až VDDIO4. Napětí 1,8 V potřebné pro jádro se odebírá přímo z vestavěného stabilizátoru na vývodu VDDOUT. Zdroj 1,8 V pro napájení PLL pak přes filtr potlačující vysoké kmitočty tvořený L1, C5 a C6.

Vnitřní PLL syntetizátor hodinového kmitočtu 84 MHz vyžaduje zdroj referenčního kmitočtu 12 MHz tvořeného oscilátorem s krystalem Q1. Kondenzátory C3 a C4 slouží pro nastartování oscilátoru po přivedení napájecího napětí. Hodnota 22 pF je zvolena na základě doporučení uvedeného v katalogovém listu výrobce.

## 12.2 Napájení

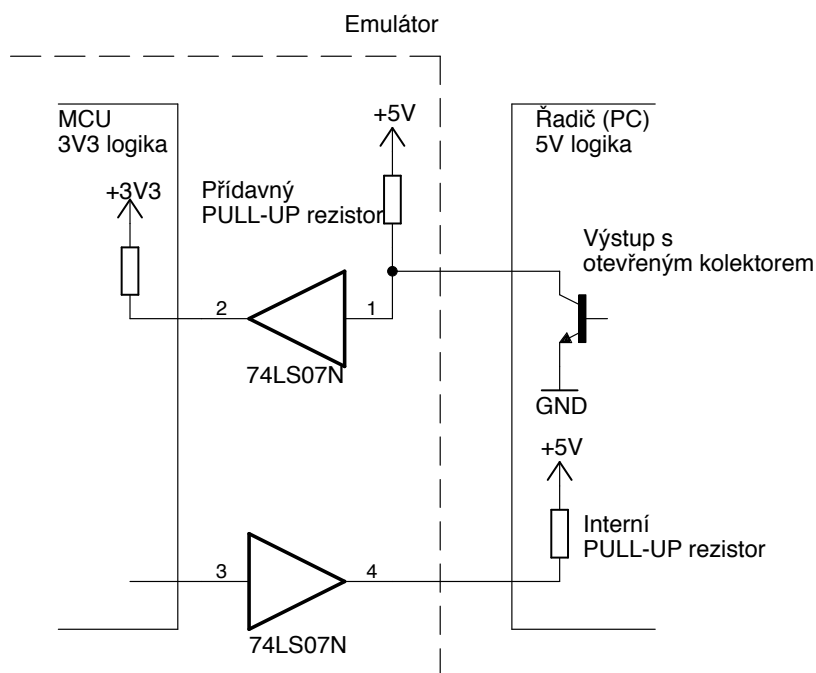
Pro činnost mikrokontroléru a SD karty je zapotřebí napětí 3,3 V. Napájení běžné disketové mechaniky je zajištěno pomocí kabelu zakončeným konektorem typu Molex, který poskytuje napětí 12 a 5 V. Využita však bude pouze větev 5 V, která může přímo napájet budiče 74LS07. Napětí 3,3 V pro mikrokontrolér a SD kartu zajistí integrovaný stabilizátor LM2937. Kondenzátory C8 až C12 jsou blokovací a musí být na desce umístěny co nejbližší k napájecím přívodům obvodů.

## 12.3 Komunikace s diskovým řadičem

Jak již bylo posáno v kapitole 4, komunikace mechaniky s řadičem probíhá celkem pomocí 15ti signálů o TTL úrovních zakončených přizpůsobovacími rezistory o hodnotě 150  $\Omega$  až 1,5 k $\Omega$  připojenými na napájecí napětí. Použitý mikrokontrolér ATSAM3X8E pracuje s napájecím napětím 3,3 V, interface mechaniky pak s 5 V. Oboustranné přizpůsobení je tedy vhodné zajistit pomocí neinvertujících budičů s otevřeným kolektorem. Vhodným obvodem ze série 74xx je

například 74LS07. Obvod 74LS07 je napájen napětím 5 V a vzhledem ke kompatibilitě napěťových úrovní 3,3 V logiky je možné jej využít. Mikrokontrolér umožňuje nastavit vnitřní pull-up rezistory na jednotlivých pinech. Na výstupech obvodů 74LS07 připojených na vstupy MCU tak není potřeba použít externí rezistory. Všechny signály jsou vyvedeny na příslušné piny interface konektoru X1.

Hodnota odporu zakončovacích rezistorů se může při použití emulátoru v různých zařízeních měnit. Z tohoto důvodu bude v zapojení použito rezistorového pole RN1 umístěného v patici. Případná výměna za jinou hodnotu či úplné odpojení v případě potřeby bude usnadněna.



Obrázek 21: Využití budičů 74LS07

## 12.4 Úložiště dat

Jako úložiště dat bude sloužit paměťová karta SD. Kromě datových souborů bude sloužit k uložení konfiguračního souboru, který obsahuje veškerá data nutná pro správnou funkci emulované mechaniky. Komunikace s kartou je sériová a může pracovat ve dvou módech.

### 12.4.1 SD mód

SD mód se vyznačuje především velkou přenosovou rychlostí s možností využití 4-bitové komunikace. Tento mód se využívá především u multimediálních zařízení s vysokými požadavky na přenosovou rychlost, která dosahuje řádově desítky MB/s. Využití pro embedded aplikace nebývá příliš časté, poněvadž vyžaduje přítomnost HSMCI periferie na mikrokontroléru.

#### 12.4.2 SPI mód

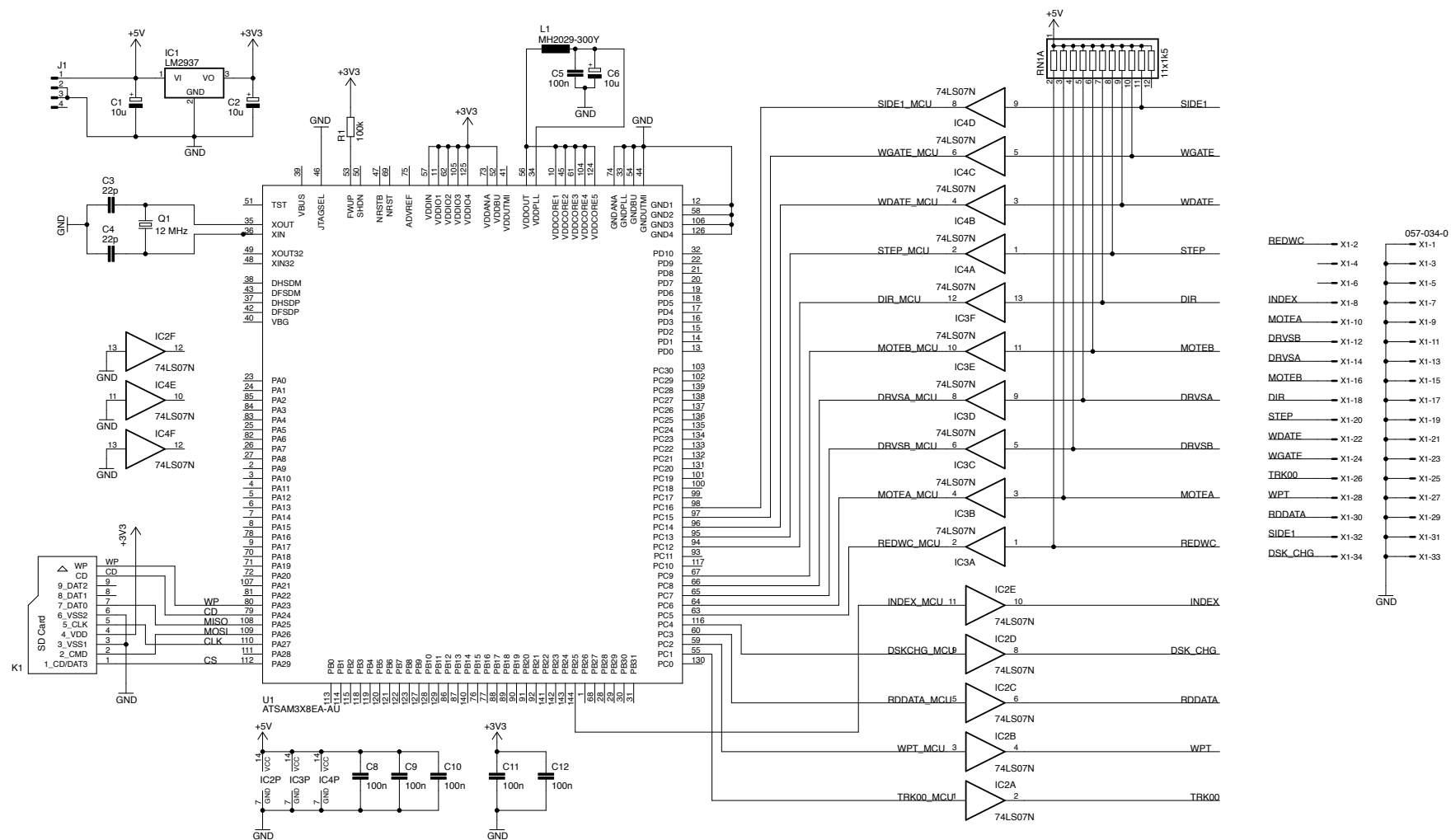
Komunikace probíhá dle standardu sběrnice SPI, kdy MCU představuje řídicí zařízení (Master) a SD karta pak podřízené zařízení (Slave). Přenosová rychlost závisí na frekvenci použitého hodinového signálu. Běžné karty umí pracovat do kmitočtu 20MHz, některé až do 25MHz. Vzhledem ke kompatibilitě a možnosti použití co nejširšího množství karet je vhodné používat taktovací kmitočet do 20MHz. Pro jednoduchost implementace a dostačující přenosovou rychlost byl pro realizaci emulátoru vybrán právě tento mód.

#### 12.4.3 Připojení paměťové karty

Karta je umístěna ve slotu K1, který umožňuje její snadnou výměnu. Komunikace pak probíhá pomocí signálů SPI sběrnice, kde signály MOSI a MISO jsou datové, CLK je hodinový signál a CD slouží k výběru SD karty. Slot obsahuje ještě mikrosvínače sloužící k detekci vložené karty (signál CD) a uzamčení karty proti zápisu (signál WP).



Obrázek 22: Prototyp emulátoru



Obrázek 23: Schéma zapojení



## 13 Obslužný program

Obslužný program představuje firmware uložený ve flash paměti mikrokontroléru. Zdrojový kód je psán v jazyce C, některé úseky vyžadující přesné časování pak v Assembleru. Program využívá pro některé funkce několik volně dostupných softwarových knihoven.

### 13.1 Ovladače

Komunikace s periferiemi mikrokontroléru probíhá přes registry. Výrobce poskytuje pro své mikrokontroléry i základní programové vybavení ve formě ovladačů, které usnadňují práci s periferiemi.

#### 13.1.1 SPI

Ovladač zajišťuje komunikaci mezi SD kartou a mikrokontrolérem. Pro zvýšení přenosové rychlosti bylo nutné stávající ovladač poskytovaný výrobcem od základu upravit, aby při každém přístupu neprováděl kontrolu správnosti nastavení jednotlivých pinů SPI sběrnice.

#### 13.1.2 UART

UART je použitý pouze pro účely ladění programu. Veškeré stavové a chybové hlášky jsou vypisovány na sériovou sběrnici, na kterou je možné se připojit a hlášky zobrazit v konzoli nastavenou na 115200 baud/s, žádná parita a 1 stop bit.

#### 13.1.3 PWM

PWM modul je využit pro generování datového signálu. Nastavení šířky pulzu na 400 ns se provede zápisem hodnoty 0x22 do registru PWM\_CDTY. Periodu je pak možné měnit zápisem konkrétních hodnot do registru periody PWM\_CPRD. Hodnoty zapisované do registru odpovídají periodám uvedeným v tabulce.

Tabulka 12: Hodnoty registru pro konkrétní hodnoty periody

Perioda signálu	Hodnota registru
2 $\mu$ s	0xA8
3 $\mu$ s	0xFC
4 $\mu$ s	0x150
6 $\mu$ s	0x1F8
8 $\mu$ s	0x2A0

### 13.1.4 TIMER

Časovače jsou v programu využity dva. První slouží ke generování průběhu signálu INDEX s periodou odpovídající nastavené rychlosti otáčení. Časovač je tedy nutné nastavit do režimu generování signálu zápisem hodnoty 0x9C001 do konfiguračního registru kanálu časovače TC\_CMR. Pro rychlost otáčení 300 ot/min je perioda 200 ms o šířce pulzu 3ms musí registr TC\_RA obsahovat hodnotu 0x7AFC a registr TC\_RC pak hodnotu 0x200AD6.

Druhý časovač slouží pro měření periody příchozích pulzů zapisovaných dat. Časovač je nastaven tak, aby uložil svou hodnotu s příchodem každé sestupné hrany signálu WDATE. Zároveň dojde k vyvolání přerušení, které vypočítá rozdíl hodnoty čítače vždy mezi dvěma pulzy. Rozdíl hodnot je přímo úměrný periodě příchozího datového signálu. Tato hodnota je následně zpracována softwarovým dekodérem a po jednotlivých bitech zapsána do bufferu stopy.

### 13.1.5 GPIO

Pro jednoduchost je využit přímý zápis do registrů, pomocí kterých je možno nastavit, zda se konkrétní pin daného portu bude chovat jako vstupní či výstupní. Pro výstupní pin je pak možné nastavit úroveň, která se na výstupu objeví. Vzhledem k použití obvodů s otevřeným kolektorem na vstupech, je nutné pro správnou funkci aktivovat vnitřní pull-up rezistory.

## 13.2 Přídavné knihovny

Ovládací program využívá pro svou činnost dvou volně dostupných knihoven pro zápis dat v souborovém systému FAT32 a pro parsování konfiguračního souboru.

### 13.2.1 FatFS

FatFS je softwarová knihovna umožňující čtení a zápis dat s využitím souborového systému FAT32. Vzhledem k jednoduchosti použití a malým nárokům na velikost programové paměti se hodí zejména pro embedded aplikace. Knihovna je psaná v programovacím jazyce C a pro svou funkci vyžaduje doprogramovat pouze fyzickou vrstvu pro komunikaci se zařízením. To obnáší implementaci následujících čtyř funkcí:

- **disk\_initialize** - Tato funkce se volá při prvním přístupu k datům v paměti. Jejím úkolem je nastavit komunikační kanál (v tomto případě SPI sběrnici) a připravit paměť na komunikaci.
- **disk\_status** - Funkce vrací stav průběhu komunikace. Využívá se pouze v případě použití neblokujícího způsobu přenosu, který není u současné implementace emulátoru použit.
- **disk\_read** - Funkce zodpovědná za čtení dat z fyzické paměti. Parametry funkce jsou buffer výstupních dat, adresa v paměti, od které má čtení probíhat a počet bajtů, které se mají přečíst.

- **disk\_write** - Obdoba funkce read pro zápis dat z fyzické paměti. Parametry funkce jsou: buffer zapisovaných dat, adresa v paměti, na kterou má zápis proběhnout a počet bajtů, které se mají zapsat.

Knihovna FatFS je distribuována jako open source, vydána pod BSD licencí, umožňující využití i v komerčních aplikacích.

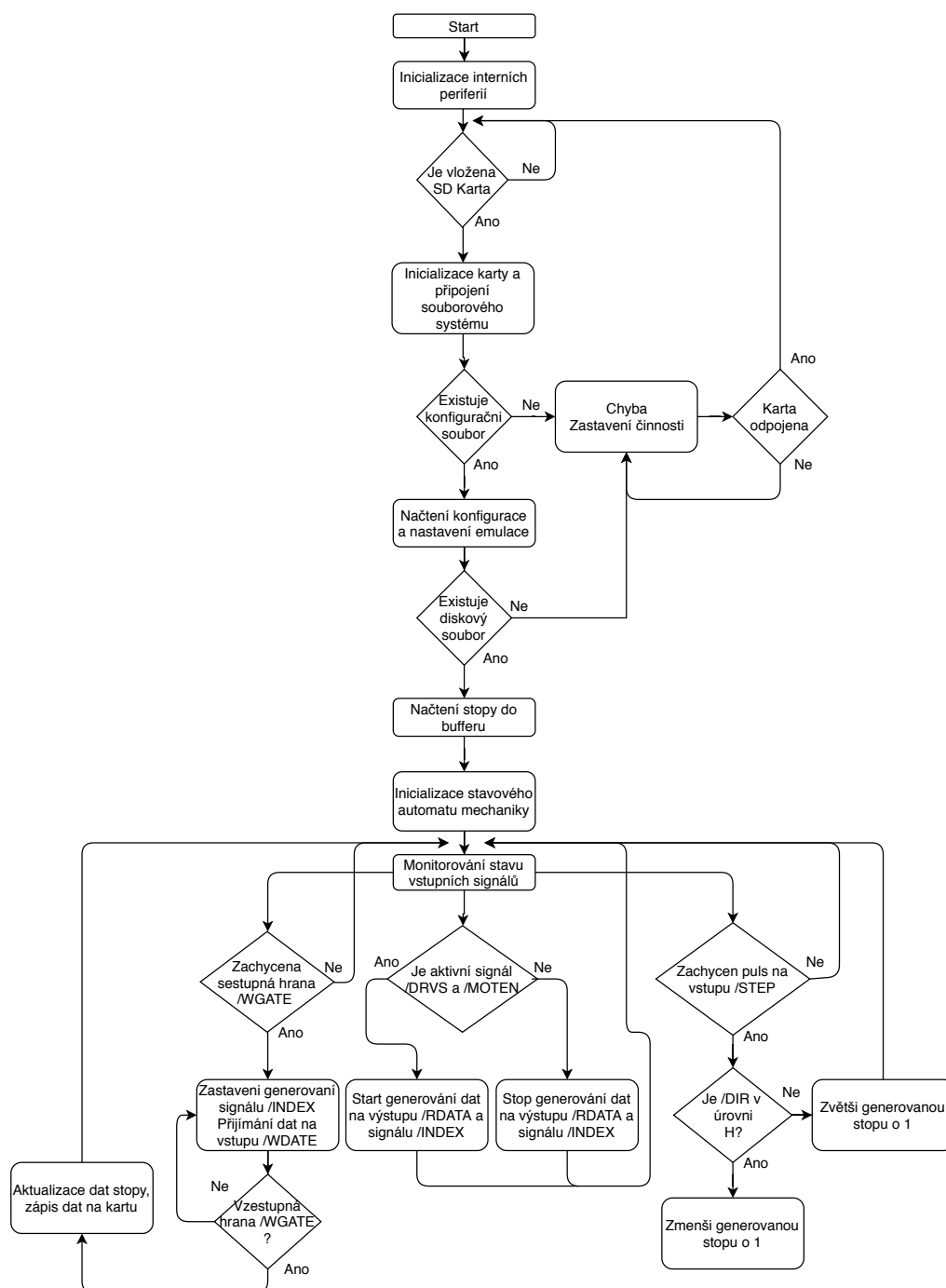
### 13.2.2 JSMN

Knihovna JSMN je využita pro čtení konfiguračních parametrů ze souboru ve formátu JSON. Je psaná v jazyce C, distribuovaná jako open source pod licencí MIT, umožňující integraci i do komerčních aplikací. Příklad nastavení konfiguračního souboru emulátoru spolu s popisem jednotlivých parametrů je uveden v příloze.

## 13.3 Aktualizace firmware mikrokontroléru

Aktualizaci firmware je možné provést bez využití programátoru přímo pomocí souboru s binárními daty zkompilovaného obslužného programu. Soubor obsahující data programu je pak nutné uložit do kořenového adresáře SD karty s názvem firmware\_update.fwu. Při zapnutí emulátoru se vždy detekuje přítomnost souboru s tímto názvem. Pokud je soubor přítomen, program ověří jeho integritu a firmware se zapíše do programové paměti mikrokontroléru a provede reset. Soubor fwu je doplněný o 4 bajty CRC kontrolního součtu na konci dat, aby se předešlo nechtěnému přepisu dat v programové paměti. Samotný přepis dat pak provádí bootloader vestavěný v ROM paměti mikrokontroléru.

### 13.4 Vývojový diagram obslužného programu



Obrázek 24: Vývojový diagram programu

## 14 Ověření funkce emulátoru

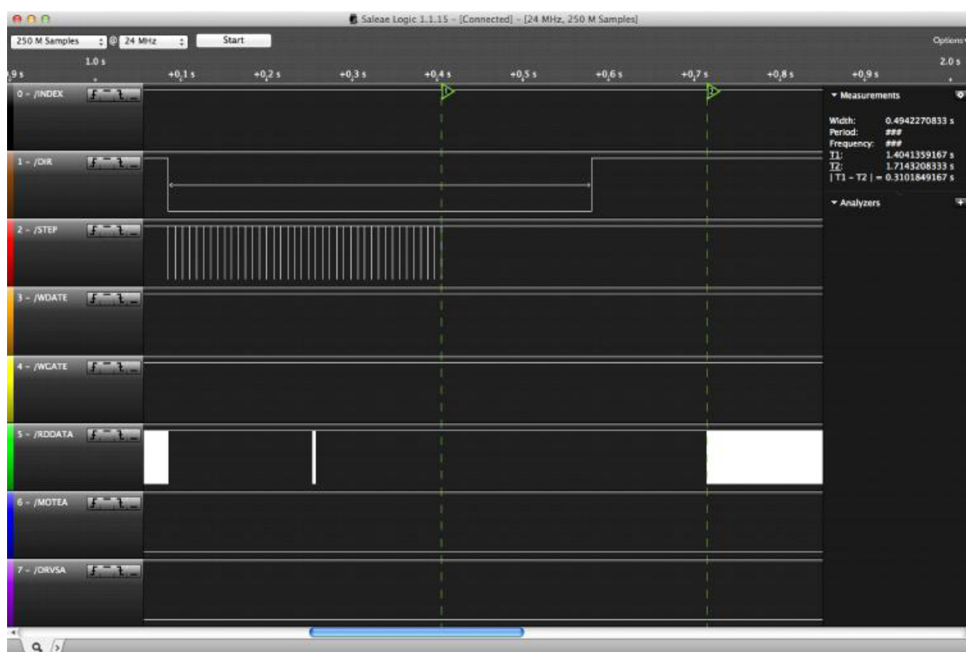
Pro ověření funkce zařízení a odladění ovládacího programu mikrokontroléru bylo celé zapojení realizováno na univerzální desce plošných spojů. Za účelem prověření kompatibility byla provedena zkouška ve třech různých zařízeních s rozdílnými požadavky na nasatavení jednotlivých parametrů.

### 14.1 Emulace 3,5" a 5,25" mechaniky v PC

K ověření funkce emulace 3,5" a 5,25" mechaniky je nejvýhodnější použít klasické PC, které disponuje integrovaným diskovým řadičem a podporuje oba typy disketových mechanik. Navíc je možné použít některou z mnoha volně dostupných diagnostických aplikací, které se zaměřují právě na testování správnosti čtení, zápisu a celkového chování mechaniky. Testování probíhalo pomocí diagnostického programu CheckIt. Emulátor byl nejprve nastaven pro emulaci 3,5" mechaniky s kapacitou 1,4 MB a standardním DOS formátem, stejné testy byly následně provedeny i pro 5,25" mechaniku s kapacitou 1,2 MB.

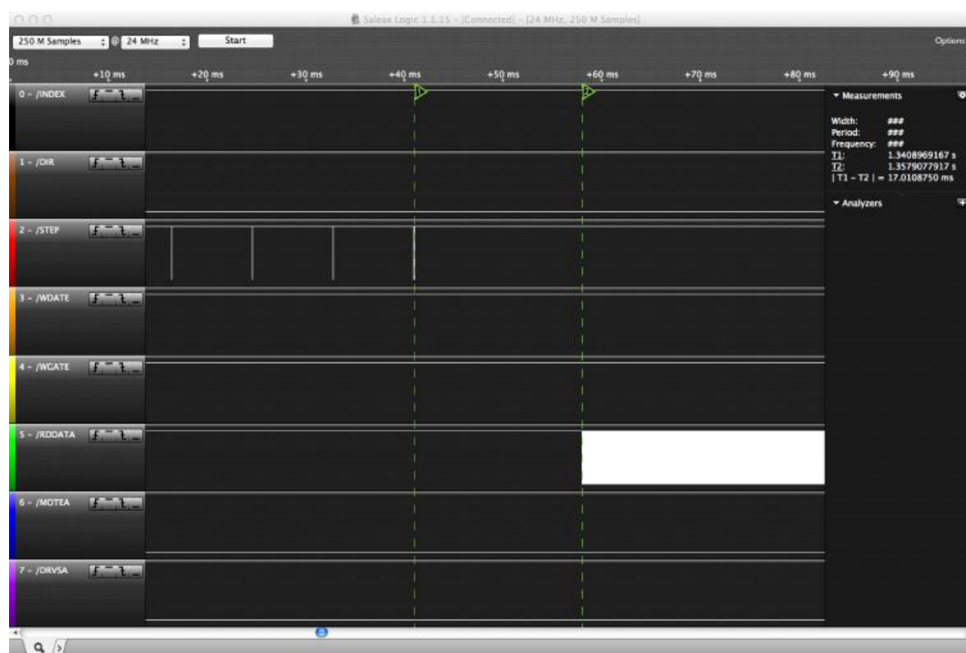
#### 14.1.1 Časování signálů při čtení dat

První fáze testování byla zaměřena na ověření správného časování jednotlivých signálů při čtení dat. Kontrola probíhala pomocí logického analyzátoru, připojeného na sběrnici mezi emulátor a řadič disketové mechaniky. Snímek průběhů signálů na obrázku 25 byl pořízen při provádění přesunu hlavy ze stopy 0 na stopu 40.



Obrázek 25: Nesprávné časování signálu /RDDATA

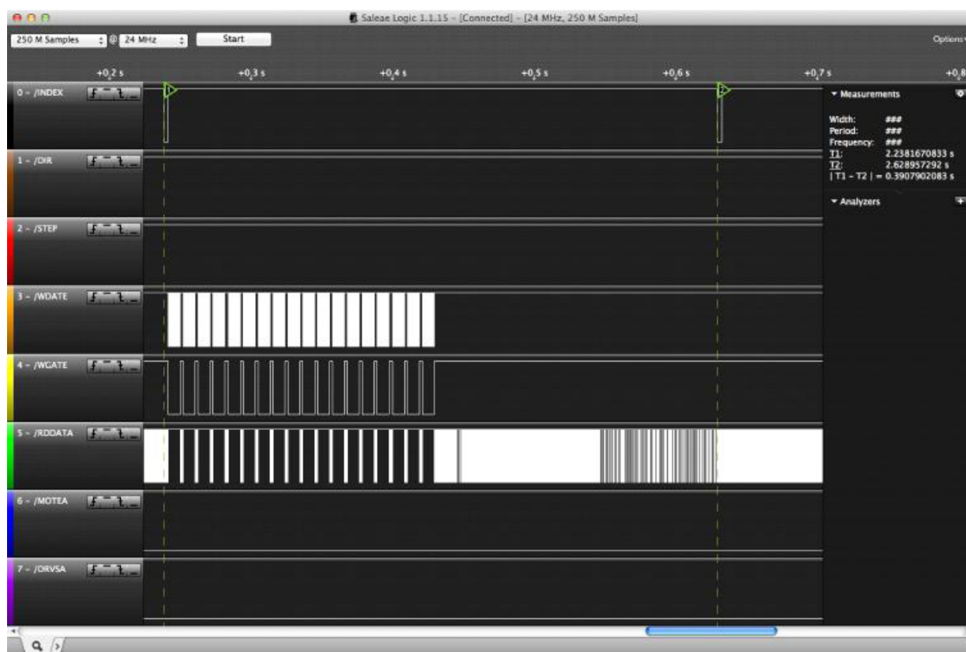
Na průběhu signálu /STEP je možné pozorovat odpovídajících 40 pulzů, které posílá řadič do emulátoru a změnu úrovně signálu /DIR na L pro pohyb hlav směrem ke středu disku. Je však zřejmé, že signál /RDDATA nesplňuje požadavky na správné časování uvedené v kapitole 4. S příchodem prvního pulzu na lince /STEP musí dojít k zablokování výstupních dat na signálu /RDDATA a jeho přechodu do úrovně H. Platná data se pak na /RDDATA musí objevit nejpozději do 17,9 ms od příchodu posledního pulzu na lince /STEP. První chybou je tedy krátký předčasný výskyt signálu /RDDATA v čase mezi 0,2 a 0,3 s a následné příliš velké zpoždění při odblokování signálu /RDDATA po příchodu posledního /STEP pulzu (310 ms místo max 17,9 ms). Příčinou chyby byl špatně zvolený postup výpočtu potřebného zpoždění blokování signálu /RDDATA v programu. Po opravě se již časy pohybují v toleranci. Viz obrázek 26.



Obrázek 26: Správné časování signálu /RDDATA

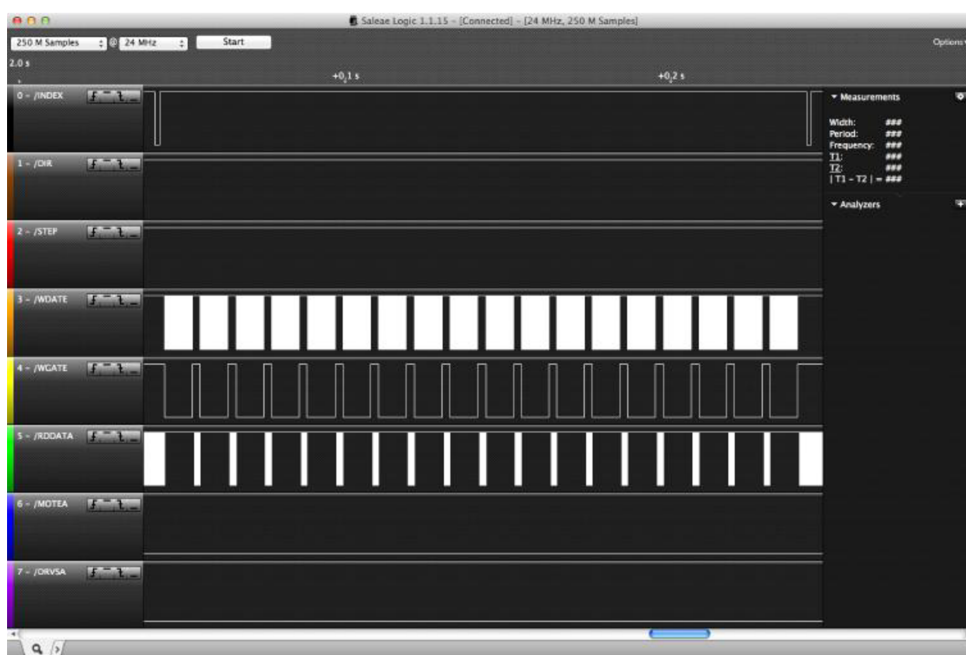
#### 14.1.2 Časování signálů při zápisu dat na disk

Dalším krokem bylo ověření správnosti signálů při zápisu dat na emulovanou disketu. Průběhy signálů na obrázku 27 zobrazují zápis všech 18ti sektorů do jedné stopy. S každým zápisem jednotlivého sektoru je možné pozorovat správnou aktivaci signálu /WGATE a současně zablokování výstupních dat na výstupu /RDDATA. Průběh však odhaluje chybu v časování /INDEX pulzů. Vlivem chyby ve stavovém automatu generování signálů, docházelo k zablokování hodinového signálu do periferie časovače při úrovni L signálu /WGATE. To následně způsobilo příliš velkou periodu signálu /INDEX a následný rozpad synchronizace, včetně poškození všech zapsaných dat.



Obrázek 27: Nesprávné časování signálu /INDEX při zápisu dat

Po nalezení a odstranění chyby již dochází ke správnému generování pulzu /INDEX těsně po ukončení zápisu posledního sektoru na emulovaný disk, viz obrázek 28.

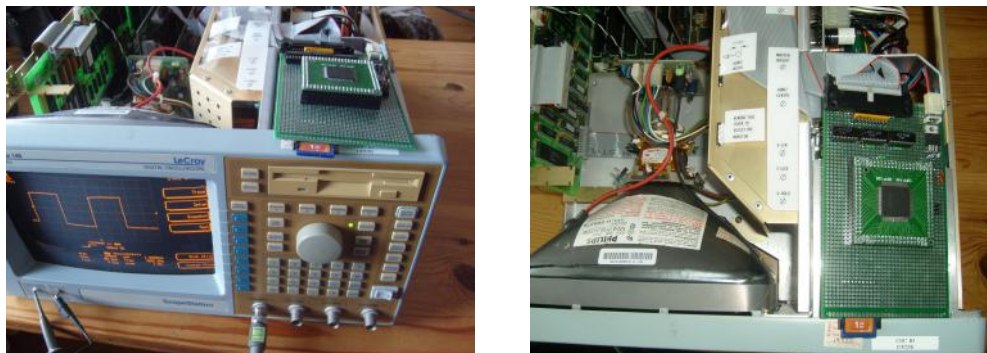


Obrázek 28: Opravené časování signálu /INDEX při zápisu dat



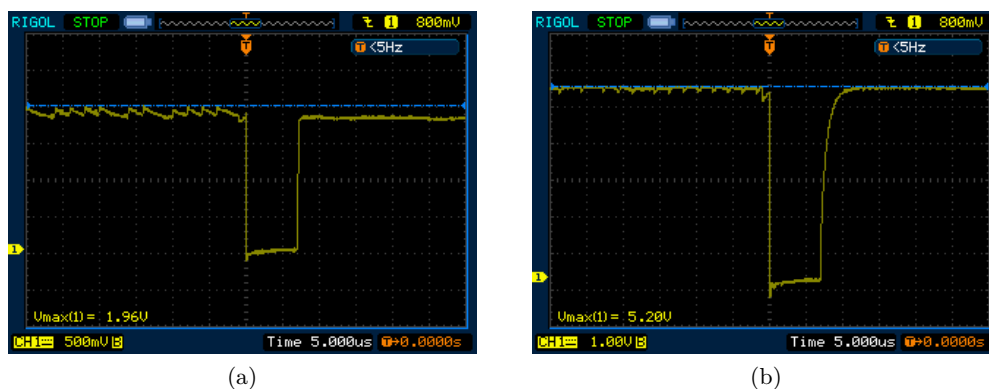


je uložení dat měřeného průběhu na emulovanou disketu, jeho opětovné načtení a zobrazení v osciloskopu a následný přenos všech dat do PC pomocí SD karty.



Obrázek 30: Náhrada mechaniky osciloskopu LeCroy

Osciloskop při prvním zapnutí s nainstalovaným emulátorem zahlásil chybu mechaniky. Po kontrole signálů na sběrnici bylo zjištěno, že napětí při úrovni H je příliš nízké a pohybuje se v zakázaném pásmu, viz obrázek 31(a). Příčinou byla příliš vysoká hodnota odporů zakončovacích (pull-up) rezistorů emulátoru. Po výměně rezistorového pole v patici emulátoru za hodnotu  $150\ \Omega$ , začal osciloskop s emulátorem správně fungovat.



Obrázek 31: Průběh signálu STEP pro (a)  $R = 1,5\ \text{k}\Omega$ , (b)  $R = 150\ \Omega$

### 14.3 Emulace 8" mechaniky v počítači Consul 2715

Správnou funkci s nastavením pro emulaci 8" mechaniky je zapotřebí ověřit na velmi starém zařízení, které tyto typy mechanik podporuje. Záměrně byl proto vybrán historický počítač Consul 2715 československé výroby, poněvadž podporuje všechny běžně používané formáty 8" disket od velikosti 250 kB až do 1 232 kB s FM i MFM kódováním. Test probíhal s využitím vestavěného testovacího programu disketové mechaniky v ROM paměti počítače. Následující test spočíval v nastartování systému z emulované systémové diskety a spuštěním aplikačního programu šachy. Program umožňuje hrát i několik hodin sám se sebou a při každém tahu současně čte i zapisuje data na disk. Lze ho tedy využít jako zátěžový test. Program hru úspěšně po 48minutách dokončil bez zamrznutí počítače.



Obrázek 32: Počítač Consul 2715

## 15 Závěr

Náplní této práce bylo se seznámit s funkcí disketových mechanik používaných ve starších přístrojích k ukládání dat a navrhnout vhodnou alternativou její náhrady za spolehlivější a trvanlivější paměťové médium. Cílem navrženého zařízení je usnadnit přenos dat do zařízení, která již disketovou mechaniku neumožňují připojit. Na základě těchto požadavků bylo navrženo zapojení emulátoru s použitím mikrokontroléru ATSAM3X8E a paměťové karty typu SD.

Zdrojový kód pro ovládací program mikrokontroléru je psán v programovacím jazyce C s požadavkem co největší možné konfigurovatelnosti všech parametrů mechaniky pomocí konfiguračního souboru. Tento soubor je uložen na paměťové kartě spolu s daty emulovaných disket. Úpravu parametrů emulovaných mechanik je tedy možné provést bez nutnosti zásahu do firmwaru mikrokontroléru. Firmware emulátoru je možné aktualizovat bez nutnosti použití dalšího hardwaru uložením nové verze programu na SD kartu. Aktualizace pak proběhne automaticky po zapnutí emulátoru. Tato koncepce zajišťuje poměrně snadné rozšíření emulátoru o další funkce a opravu případných chyb v programu.

Odladění programu a ověření celkové funkce emulátoru v reálném zařízení bylo provedeno na třech různých zařízeních, která vyžadují zcela odlišné parametry nastavení. První fáze testování probíhala na klasickém PC, za účelem ověření a odladění kódovacích a dekodovacích algoritmů. K ověření bylo využito volně dostupných diagnostických programů v prostředí systému DOS. Testována byla především správnost čtených a zapisovaných dat náhodného charakteru a také časování výstupních signálů. Po odstranění všech nalezených chyb a nedostatků v programu se podařilo dosáhnout spolehlivé funkce a bezchybné procházení testů pro náhodné čtení i zápis při emulaci 3,5" i 5,25" mechaniky s kódováním MFM.

Cílem druhé fáze testů byla kontrola kompatibility emulátoru s dalším typem diskového řadiče. Test potvrdil nutnost použití vyměnitelného rezistorového pole pro funkci zakončovacích odporů.

Třetí fáze testů měla za úkol prověřit celkovou funkci emulátoru pro různá nastavení konfiguračního souboru a současnou emulaci dvou mechanik současně. Jako testovací zařízení v tomto případě posloužil počítač Consul 2715, který může pracovat s 8" mechanikami a podporuje všechny běžně používané formáty 8" disket od velikosti 250 kB až po 1 232 kB s FM i MFM kódováním. Emulátor prošel 48minutovým zátěžovým testem při náhodném čtení i zápisu.

Všechny zjištěné problémy, které se při testování vyskytly se podařilo v ovládacím programu opravit. Emulátor by tak měl vyhovět jako plnohodnotná náhrada 3,5", 5,25" a 8" disketové mechaniky v zařízeních, využívajících standartní interface k jejímu připojení.

## Literatura

- [1] MANN, B. *C pro mikrokontroléry*. Praha: nakladatelství BEN - Technická Literatura, 2003. 280 s. ISBN 80-7300-077-6.
- [2] ATMEL CORPORATION, *SAM3X / SAM3A Series Data Sheet*, katalogový list, 2015, 459s. Dostupné z: <[http://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a\\_datasheet.pdf](http://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a_datasheet.pdf)>.
- [3] STANDARD MICROSYSTEMS CORPORATION, *FDC 9216 Floppy Disk Data Separator*, katalogový list, 1982, 12s. Dostupné z: <[http://www.bitsavers.org/components/standardMicrosystems/TN6-1\\_9216\\_Floppy\\_Disk\\_Data\\_Separator\\_Jun82.pdf](http://www.bitsavers.org/components/standardMicrosystems/TN6-1_9216_Floppy_Disk_Data_Separator_Jun82.pdf)>.
- [4] TEAC, *TEAC FD-05HF-8830 SPECIFICATION*, katalogový list, 2005, 29. Dostupné z: <<https://hxc2001.com/download/datasheet/floppy/thirdparty/Teac/TEAC%20FD-05HF-8830.pdf>>.
- [5] GoTek 2019 [cit. 2. 3. 2019]. *GoTek USB Floppy Emulator* [online]. Dostupné z URL: <<http://www.gotekemulator.com>>.
- [6] Solid State Disks Ltd 2019 [cit. 2. 3. 2019]. *FloppyFlash Emulator* [online]. Dostupné z URL: <<https://solidstatedisks.co.uk/index.php/products/floppy-scsi-ssd-drives/floppyflash-floppy-drive-emulations-1/floppyflash-emulator-to-replace-legacy-floppy-disk-drives-detail>>.
- [7] HxC 2001 [cit. 2. 3. 2019]. *HxC Floppy Emulator* [online]. Dostupné z URL: <[http://hxc2001.free.fr/floppy\\_drive\\_emulator/](http://hxc2001.free.fr/floppy_drive_emulator/)>.
- [8] Wikipedia [cit. 12. 2. 2019]. *List of floppy disk formats* [online]. Dostupné z URL: <[https://en.wikipedia.org/wiki/List\\_of\\_floppy\\_disk\\_formats](https://en.wikipedia.org/wiki/List_of_floppy_disk_formats)>.
- [9] Wikipedia [cit. 8. 2. 2019]. *Floppy-disk controller* [online]. Dostupné z URL: <[https://en.wikipedia.org/wiki/Floppy-disk\\_controller](https://en.wikipedia.org/wiki/Floppy-disk_controller)>.
- [10] Motorola Semiconductor, *MC3470A*, katalogový list, 1984, 14s. Dostupné z: <[http://bitsavers.trailing-edge.com/components/motorola/\\_dataSheets/MC3470.pdf](http://bitsavers.trailing-edge.com/components/motorola/_dataSheets/MC3470.pdf)>.
- [11] Old pinouts [cit. 1. 3. 2019]. *Floppy Diskdrive pinout* [online]. Dostupné z URL: <[http://old.pinouts.ru/HD/InternalDisk\\_pinout.shtml](http://old.pinouts.ru/HD/InternalDisk_pinout.shtml)>.
- [12] shoreline-electronic [cit. 16. 3. 2019]. *3.5 inch drive* [online]. Dostupné z URL: <<http://www.shoreline-electronics.com/1%20inch%20floppy%20drives.htm>>.
- [13] Wikipedia [cit. 3. 4. 2019]. *Cyklický redundantní součet* [online]. Dostupné z URL: <[https://cs.wikipedia.org/wiki/Cyklický\\_redundantní\\_součet](https://cs.wikipedia.org/wiki/Cyklický_redundantní_součet)>.

- [14] HAARDT, M, KNAFF, A. NIEMI, D. *The floppy user guide*, Online dokument, 2001, 26s. Dostupné z: <<http://www.hermannseib.com/documents/floppy.pdf>>.
- [15] General Floppy Drives information [cit. 20. 4. 2019]. *Atari ST FD Hardware* [online]. Dostupné z URL: <<https://info-coach.fr/atari/hardware/FD-Hard.php>>.
- [16] KHOL, I. *Diskety a disketové jednotky* Praha: Amaro spol. s r.o - Technická Literatura, 1989. ISBN 0322-9572.
- [17] INTEL CORPORATION, *82077A Floppy disk controller*, katalogový list, 1996, 62s. Dostupné z: <<http://www.buchty.net/casio/files/82077.pdf>>.

## A Příloha

### A.1 Hlavní smyčka emulačního programu

---

```
status_t FLOPPY_main()
{
    FATFS fs;
    FRESULT result;
    int bytesWrite= 0;
    uint8_t actTrack = 0;
    uint8_t actSide = 0;
    uint8_t shift = 0;
    bool update = 0;
    memset(dat, 0, sizeof(dat));

    /* Inicializace hardware pro generovani signalu INDEX, RDDATA, atd */
    FLOPPY_IndexInit();
    FLOPPY_WrDataInit();
    FLOPPY_PwmInit();
    FLOPPY_ConfigureOutputs();
    PIOA->PIO_PER |= 0x1;
    PIOA->PIO_OER |= 0x1;

    /* Pripojeni souboroveho systemu FAT na SD karte */
    result = f_mount(&fs,"1:", 1);
    if (result != FR_OK)
    {
        PRINTF("Chyba pri pripojeni disku!\n");
        return kStatus_Fatal;
    }

    /* Otevreni souboru obrazu disku pro emulaci */
    result = f_open(&file, g_filename, FA_READ | FA_WRITE);
    if (result != FR_OK)
    {
        PRINTF("Chyba pri otevirani souboru!\n");
        return kStatus_Fatal;
    }
    else
```

```

{
    PRINTF("Soubor byl uspesne otevren.\n");
}

/* Nacteni dat stopy a prekodovani do FM/MFM */
FLOPPY_LoadTrack(g_track, g_side);
FLOPPY_ProcessTrack(g_track, g_side);

/* Hlavni emulacni smycka */
while(1)
{
    /* Dekodovani prichozich dat a synchronizacni sekvence */
    while((wrIdx > procIdx) && sectToWr)
    {
        data_decoder((wrData[procIdx]) & 3);
        data_decoder((wrData[procIdx] >> 2) & 3);
        data_decoder((wrData[procIdx] >> 4) & 3);
        data_decoder((wrData[procIdx] >> 6) & 3);
        if(procCount > 512)
        {
            procCount=0;
            sectToWr--;
            g_synched = 0;
        }
        procIdx++;
        update = 1;
    }
    /* Propsani aktualnich dat stopy v RAM do souboru */
    if((sectToWr == 0) && update)
    {
        update = 0;
        procIdx = 0;
        sectToWr = 0;
        wrIdx = 0;
        shift = 0;
        g_idx = 0;
        g_synched = 0;
        g_shift = 0;
        for(char x = 0; x<sect_cnt; x++)

```

```

{
    if(sectToWrite[x] == 0)
    {
        break;
    }
    memcpy(&trackData[(sectToWrite[x]-1)*sect_size], &dat[(x*sect_size)
        +(x+1)], sect_size);
}
sectToWriteIdx = 0;
memset(dat, 0, sizeof(dat));
memset(sectToWrite, 0, sizeof(sectToWrite));
memset(wrData, 0, sizeof(wrData));
FLOPPY_ProcessTrack(g_track, g_side);
}
/* Nacteni novych dat stopy pri prechodu na novou stopu */
if((actTrack != g_track) || (g_side != actSide))
{
    myDelay(); //Zpozdeni pri pri posunu hlav na danou stopu
    result = f_lseek(&file, (actTrack * (sect_size * sect_cnt * 2)) + (
        actSide * (sect_size * sect_cnt)));
    if(result != FR_OK)
    {
        PRINTF("Chyba pri f_lseek!\n");
        return kStatus_Fatal;
    }
    for(char i=0; i < sect_cnt; i++)
    {
        result = f_write(&file, &trackData[i*sect_size], sect_size, &
            bytesWrite);
        if ((result != FR_OK) || (bytesWrite != sect_size))
        {
            PRINTF("Chyba zapisu do souboru!\n");
            return kStatus_Fatal;
        }
    }
}
actTrack = g_track;
actSide = g_side;
FLOPPY_LoadTrack(g_track, g_side);
FLOPPY_ProcessTrack(g_track, g_side);

```



```

    }
}
}

```

---

Výpis 2: Hlavní programová smyčka

## A.2 Příklad úplného nastavení v souboru floppy.cfg

---

```

{
    "pocet_mechanik": 1,
    "nazev_souboru_mech_A": "DISK_A.IMG",
    "nazev_souboru_mech_B": "DISK_B.IMG",
},
{
    "kodovani_mech_A": "MFM"
    "hustota_zazn_mech_A": "high"
    "pocet_stran_mech_A": 2
    "pocet_stop_mech_A": 80
    "pocet_sektoru_mech_A": 18
    "velikost_sektoru_mech_A": 512
    "start_synch_pattern_mech_A": [ 0xC2, 0xC2, 0xC2 ]
    "filler_znacka_mech_A": "0x4E"
    "index_znacka_mech_A": "0xFC"
    "hlavicka_znacka_mech_A": "0xFE"
    "data_znacka_mech_A": "0xFB"
    "sektor_synch_pattern_mech_A": [ 0xA1, 0xA1, 0xA1 ]
},
{
    "kodovani_mech_B": "MFM"
    "hustota_zazn_mech_B": "double"
    "pocet_stran_mech_B": 2
    "pocet_stop_mech_B": 40
    "pocet_sektoru_mech_B": 9
    "velikost_sektoru_mech_B": 512
    "start_synch_pattern_mech_B": [ 0xC2, 0xC2, 0xC2 ]
    "filler_znacka_mech_B": "0x4E"
    "index_znacka_mech_B": "0xFC"
    "hlavicka_znacka_mech_B": "0xFE"
}

```

```

"data_znacka_mech_B": "0xFB"
"sektor_synch_pattern_mech_B": [ 0xA1, 0xA1, 0xA1 ]
}

```

---

Výpis 3: Příklad kompletního nastavení emulovaných jednotek souboru floppy.cfg

### A.3 Popis parametrů konfiguračního souboru

Tabulka 13: Popis parametrů - X nahradit "A" nebo "B"

Parametr	Popis	Hodnoty
pocet_mechanik	počet emulovaných mechanik	1, 2
nazev_souboru_mech_X	název souboru disku mechaniky X	<nazev_souboru>
kodovani_mech_X	použitý typ kódování dat mechaniky	"FM", "MFM"
hustota_zazn_mech_X	hustota záznamu dat	"single", "double", "high"
pocet_stop_mech_X	celkový počet stop	1 - 255
pocet_sektoru_mech_X	počet sektorů ve stopě	1 - 255
velikost_sektoru_mech_X	velikost sektoru	128 - 1024
start_synch_pattern_mech_X	synchronizační data začátku stopy	[<B1>, <B2>, <B3>]
filler_znacka_mech_X	synchronizační vyplňovací bajt	<B1>
index_znacka_mech_X	ID bajt začátku sektoru	<B1>
hlavicka_znacka_mech_X	ID bajt hlavičky sektoru	<B1>
data_znacka_mech_X	ID bajt začátku dat	<B1>
sektor_synch_pattern_mech_X	synchronizační data začátku sektoru	[<B1>, <B2>, <B3>]

## B Příloha

### B.1 Seznam použitých součástek

Tabulka 14: Seznam součástek

Označení	Hodnota	Typ	Pouzdro
C1, C2, C6	10uF/16V	Elektrolytický	
C3, C4	22pF	Keramický	
C5, C8 - C12	100nF	Keramický	
IC1	LM2937	Lin. stabilizátor	TO220
IC2 - IC4	74LS07N	Budič open-collector	DIL16
J1	4-pin	konektor Mini-Molex	4-pin Mini-Molex
K1	34-pin	MLW	4-pin MLW-34G
L1	MH2029-300Y	MH2029	Feritová perle
Q1	12 MHz	Krystal	HC49U
R1	100k	0,25W	
RN1	11 x 1,5k	CTS753DRT24	CTS753DRT24
U1	ATSAM3X8EA-AU	ATSAM3X/AAU	LQFP144